ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

# Program Self-Study Report

## For The

## Rose-Hulman Institute of Technology

## Computer Science Program

### in the Department of

### Computer Science and Software Engineering

June 26, 2006

***Prepared For:***

**Computing Accreditation Commission**
Accreditation Board for Engineering and Technology, Inc.
111 Market Place, Suite 1050
Baltimore, Maryland 21202-4012
Phone: 410-347-7700
Fax: 410-625-2238
email: cac@abet.org
URL: http://www.abet.org/

***Prepared By:***

**Rose-Hulman Institute of Technology**
Computer Science and Software Engineering Department
5500 Wabash Avenue
Terre Haute, IN 47803
Phone: 812-877-8402
Fax: 812-872-6060
URL: http://www.cs.rose-hulman.edu/

# Table of Contents


Table of Contents ..... i

List of Figures ..... v

List of Tables ..... vii

Background Information ..... 1

1. Degree Title ..... 1
2. Program Modes ..... 1
3. Actions to Correct Previous Concerns ..... 1
4. Contact Information ..... 1

Accreditation Summary ..... 3

1. Objectives, Outcomes, and Assessment ..... 3
   - A. Objectives and Outcomes ..... 4
   - B. Student Attributes: ..... 13
   - C. Assessments and Program Improvement ..... 15
   - D. Documentation ..... 38
2. Student Support ..... 39
   - A. Frequency of Course Offerings ..... 39
   - B. Interaction with Faculty ..... 39
   - C. Student Advising ..... 40
   - D. Meeting the Requirements ..... 41
3. Faculty Qualifications ..... 43
   - A. Faculty Profile ..... 43
   - B. Information Regarding Faculty Members ..... 44
4. Faculty Size and Workload ..... 89
   - A. Faculty Size ..... 89
   - B. Faculty Workload ..... 89
   - C. Program Development and Delivery ..... 89
   - D. Course Oversight ..... 90
   - E. Quality of Faculty Advising ..... 90
5. Curriculum ..... 91
   - A. Title of Degree Program ..... 91
   - B. Credit Hour Definition ..... 91
   - C. Prerequisite Flow Chart ..... 91
   - D. Course Requirements of Curriculum (term by term and year by year) ..... 94
   - E. Course Descriptions ..... 99

6. Technology Infrastructure ..... 195
A. Computing Resources ..... 195
B. Student Access ..... 198
C. Faculty Access ..... 198
D. Support Personnel ..... 198
7. Institutional Support and Financial Resources ..... 199
A. Faculty Stability ..... 199
B. Faculty Professional Activities ..... 199
C. Administration Effectiveness ..... 200
D. Adequacy of Resources ..... 200
E. Laboratory and Computing Resources ..... 201
F. Continuity of Institutional Support ..... 201
8. Institutional Facilities ..... 203
A. Library Staffing ..... 203
B. Library Technical Collection ..... 203
C. Library Electronic Access ..... 203
D. Classroom Equipment ..... 203
E. Faculty Offices ..... 204
Appendix I. Information Relative to the Entire Institution ..... 207
A. General Information ..... 207
B. Type of Control ..... 207
C. Regional or Institutional Accreditation ..... 207
D. Enrollment ..... 212
E. Promotion and Faculty Tenure ..... 212
Appendix II. General Information on the Unit Responsible for the Program ..... 215
A. Program Unit ..... 215
B. Administrative Head of Program Unit ..... 215
C. Organization Chart ..... 216
D. Computer-Related Undergraduate Degree Programs ..... 217
Appendix III. Finances ..... 219
A. Finances Related to the Program(s) ..... 219
B. Operating and Computing Expenditures for the Five Fiscal Years Immediately Preceding that Reported in III A ..... 220
C. Additional Funding ..... 220
Appendix IV. Program Personnel and Policies Towards Consulting, Professional Development, and Recruiting ..... 221

A. Term of Appointment of Administrative Head .....221
B. Number of Personnel Associated with Program.....221
C. Policies .....221
Appendix V. Program Enrollment and Degree Data.....223
Appendix VI. Admission Requirements.....225
A. Admission of Students.....225
Appendix VII. Program Assessment Data.....227

# List of Figures

Figure 1. Elements of CSSE Continuous Improvement Process that are relevant to Computer Science Program Outcomes. 16

Figure 2. Elements of CSSE Continuous Improvement Process that are relevant to Computer Science Program Educational Objectives. 17

Figure 3. Prerequisite relationships of selected CSSE and discrete mathematics courses. 92

Figure 4. Prerequisite relationships of other courses required for the computer science program. 93

Figure 5. Improvement in the quality of instructional equipment at Rose-Hulman since commitment to reach $1,000,000 yearly allocation. 204

Figure 6. Most recent accreditation action of North Central Association of Colleges and Schools (page 1 of 4). 208

Figure 7. Most recent accreditation action of North Central Association of Colleges and Schools (page 2 of 4). 209

Figure 8. Most recent accreditation action of North Central Association of Colleges and Schools (page 3 of 4). 210

Figure 9. Most recent accreditation action of North Central Association of Colleges and Schools (page 4 of 4). 211

Figure 10. Organizational chart illustrating direct supervisory relationship between Dean of Academic Affairs and department heads. 216

Figure 11. Organizational chart illustrating direct supervisory relationship between President's position and Dean of Academic Affairs. 217

# List of Tables

Table 1. Alignment of Program Educational Objectives with RHIT Mission Statement. 9

Table 2. Mechanisms used in the assessment and evaluation of program outcomes. 12

Table 3. Alignment of Program Outcomes and Program Educational Objectives. 13

Table 4. Alignment of CAC attributes with Program Outcomes and Institutional Learning Outcomes. 14

Table 5. REPS rating cycle. 20

Table 6. Faculty raters for Summer 2005 Rating Session. 22

Table 7. Aspects of Continuous Improvement Process most relevant to CAC accreditation criteria. 34

Table 8. Documentation of PEOs, POs, CIP, and assessment reports. 38

Table 9. CSSE courses that can be used to fulfill elective requirements of the Computer Science program. 39

Table 10. Faculty profile (name, academic rank, Ph.D. field, research areas, and advisee count)44

Table 11. Course Requirements (Freshman Year) 94

Table 12. Course Requirements (Sophomore Year) 95

Table 13. Course Requirements (Junior Year) 96

Table 14. Course Requirements (Senior Year) 97

Table 15. Placement of active fiber connections. 196

Table 16. Total number of full-time faculty, resignations, retirements, and new hires for each of the last five years. 199

Table 17. Computer-related undergraduate degree programs offered by Rose-Hulman. 218

Table 18. Financed related to the Computer Science program. 219

Table 19. Operating expenses for the CSSE department. 220

Table 20. Computer hardware/software capital expenditures (excluding equipment used primarily for research) for the CSSE department. 220

Table 21. Number of personnel associated with the Computer Science program. 221

Table 22. Five year history of enrollment and degrees conferred (Rose-Hulman as a whole). 223

Table 23. Five year history of enrollment and degrees conferred (CSSE Department). 224

Table 24. Five year history of enrollment and degrees conferred (Computer Science program). 224

# Background Information

## 1. Degree Title

Bachelor of Science in Computer Science

## 2. Program Modes

The computer science program is offered on-campus for full-time students. All classes are conducted on campus and are taught in face-to-face mode. Students have opportunities for co-op and internship experiences, but neither is required for graduation.

## 3. Actions to Correct Previous Concerns

This is the first accreditation review of the computer science program.

## 4. Contact Information

Dr. Cary Laxer, Professor and Head

Department of Computer Science and Software Engineering

Rose-Hulman Institute of Technology

5500 Wabash Avenue, CM-100

Terre Haute IN 47803-3999

812-877-8429

laxer@rose-hulman.edu

Dr. Larry Merkle, Assistant Professor of Computer Science and Software Engineering

Department of Computer Science and Software Engineering

Rose-Hulman Institute of Technology

5500 Wabash Avenue, CM-103

Terre Haute IN 47803-3999

812-877-8474

Larry.Merkle@rose-hulman.edu

# Accreditation Summary

## 1. Objectives, Outcomes, and Assessment

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

The program has documented measurable educational objectives and outcomes, based on the needs of the program's constituencies. The program uses a documented process incorporating relevant data to regularly assess its educational objectives and outcomes and to evaluate the extent to which they are being met. The results of the evaluations are used to develop and implement plans to effect continuous improvement of the program.

The program enables students to achieve the following attributes by the time of graduation:

| | |
|---|---|
| CAC-CS-a | An ability to apply knowledge of computing and mathematics appropriate to the discipline; |
| CAC-CS-b | An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution; |
| CAC-CS-c | An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs; |
| CAC-CS-d | An ability to function effectively on teams to accomplish a common goal; |
| CAC-CS-e | An understanding of professional, ethical and social responsibilities; |
| CAC-CS-f | An ability to communicate effectively with a range of audiences; |
| CAC-CS-g | An ability to analyze the impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues; |
| CAC-CS-h | Recognition of the need for, and an ability to engage in, continuing professional development; |
| CAC-CS-i | An ability to use current techniques, skills, and tools necessary for computing practices. |
| CAC-CS-j | An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices; |
| CAC-CS-k | An ability to apply design and development principles in the construction of software systems of varying complexity. |

In order to provide this evidence, this report addresses four sets of questions, in turn[1]:

1. Objectives and Outcomes. Does the program have measurable educational objectives and outcomes? Are they based on the needs of the program's constituencies?
2. Student Attributes. Does the program enable students to achieve the attributes listed above by the time of graduation?
3. Assessments and Program Improvements. Does the program use a process incorporating relevant data to regularly assess its educational objectives and outcomes, and to evaluate the extent to which they are being met? Are the results of the evaluations used to develop and implement plans to effect continuous improvement of the program?
4. Documentation. Is there appropriate documentation of the program educational objectives, the program outcomes, and the assessment and continuous improvement processes?

## A. Objectives and Outcomes

The purpose of this section is to address the following questions:

> Does the program have measurable educational objectives and outcomes? Are they based on the needs of the program's constituencies?

As such, it presents and discusses the following[2]:

- Rose-Hulman's mission and vision statements, including a response to the institutional concern remaining from the 2000 ABET EAC report.
- Rose-Hulman's Institutional Learning Outcomes[3] (ILOs), as they apply to the computer science program.
- The computer science program's program educational objectives[4] (PEOs).

---

[1] This organization, and to a large extent the organization of the answers to each of these four questions, is driven by the program's status as a pilot program for the proposed CAC criteria during the 2006-07 review cycle, and the strong preference indicated by CAC that pilot program self-studies follow the organization of the Self-Study Questionnaire dated 9/26/05. Unfortunately, this does not always lead to information being presented in the most intuitive order.

[2] In the order mandated by the self-study questionnaire, not the order given in this overview.

[3] Most documentation refers to these as "Institutional Learning Objectives." However, in preparing for the Fall 2006 ABET accreditation visit, it was determined that they align more closely with the ABET EAC and CAC notions of "outcomes." As such, some members of the Rose-Hulman community have begun referring to them as "ILOs," and this report follows that trend.

[4] As defined in the ABET CAC Self-Study Questionnaire dated 9/26/05, and as used in this report, program educational objectives are broad statements that describe the career and professional accomplishments that the program is preparing students to achieve.

- The computer science program's program outcomes[5] (POs).
- The computer science program's stakeholders.
- Relationships among Rose-Hulman's mission, the ILOs, the PEOs, the POs, and the stakeholders' needs.
- The Continuous Improvement Process (CIP) used by the department of Computer Science and Software Engineering to assess, evaluate, and continuously improve its computer science and software engineering programs..
- Recent results of the assessment and evaluation processes, including improvements that have been identified, timelines for their implementation, and their documentation..

***Mission Statement***

The mission of the Rose-Hulman Institute of Technology is

> to provide students with the world's best undergraduate education in engineering mathematics and science in an environment of individual attention and support..

Our vision is

> to be the best undergraduate engineering mathematics and science educational institution in the world.

The mission and vision are modeled by the ILOs and their associated performance criteria. The first six are common across all programs:

RH-1 (Ethics) A recognition of ethical and professional responsibilities. Performance criteria:

1. Demonstrate knowledge of a professional code of ethics.
2. Evaluate the ethical dimensions of professional engineering, mathematical, and scientific practices.

RH-2 (Contemporary Issues) An understanding of how contemporary issues shape and are shaped by mathematics, science, & engineering. Performance criteria:

1. Demonstrate an awareness of how the problem is affected by social concerns and trends.
2. Demonstrate an awareness of how the proposed solution(s) will affect culture and the environment.

RH-3 (Global) An ability to recognize the role of professionals in the global society. Performance criteria:

[5] As defined in the ABET CAC Self-Study Questionnaire dated 9/26/05, and as used in this report, program outcomes relate to the skills, knowledge, and behaviors that students acquire in their matriculation through the program.

1. Demonstrate an awareness of the historical development of cultures and societies.

2. Show an awareness of the relationships of nations and the interdependence of peoples around the globe.

RH-4 (Culture) An understanding of diverse cultural and humanistic traditions. Performance criteria:

1. Perform, interpret, analyze, or otherwise engage in artistic, literary, and/or other forms of culture.

2. Interpret, analyze, or evaluate aspects of a society other than the student's own.

RH-5 (Teams) An ability to work effectively in teams. Performance criteria:

1. Share responsibilities and duties.

2. Analyze ideas objectively to discern feasible solutions and then build consensus.

3. Develop a strategy for action.

RH-6 (Communication) An ability to communicate effectively in oral, written, graphical, and visual forms. Performance criteria:

1. Identify readers/audience, assess their previous knowledge & information needs, and organize/design information to meet the needs.

2. Provide content that is factually correct, supported with evidence, explained with sufficient detail, and properly documented.

3. Test readers/audience response to determine how well ideas have been relayed.

4. Submit work with a minimum of errors in spelling, punctuation, grammar, and usage.

It is also believed that all students should be able to demonstrate characteristics in the general categories of problem solving, interpreting data, experiments, and design, with the specific ILO's and associated performance indicators to be decided by each technical department: For the Department of Computer Science and Software Engineering (and both degree programs offered by it), these characteristics are modeled by the following ILO's[6]:

RH-CSSE-7 (Problem Solving) An ability to apply the skills and knowledge necessary for mathematical, scientific, and engineering practices. Performance criteria:

1. Given a specific implementation of a programming problem, analyze the run time of that implementation.

2. Design an algorithm to solve a problem efficiently

[6] The ILOs are numbered consecutively for readability (and document maintainability), although this means that the numbering of the Experiments and Design ILOs do not adhere to the standard Rose-Hulman numbering scheme.

3. Given a detailed specification of a programming problem, implement that specification in an appropriate programming language correctly.

4. Given a proposed implementation of a detailed specification of a programming problem, unit-test that proposed solution in an effective manner.

RH-CSSE-8 (Experiments) An ability to conduct experiments and interpret data from experiments. Performance criteria:

1. Use appropriate experimental techniques and procedures.

2. Keep detailed and accurate records of experimental procedures and data collected.

3. Estimate experimental uncertainties and use appropriate error propagation techniques.

4. Draw reasonable conclusions from results.

RH-CSSE-9 (Design) An ability to design a product or process to satisfy a client's needs subject to constraints. Performance criteria:

1. Develop a design specification that addresses customer/client needs and constraints.

2. Carry out a conceptual design by generating multiple solutions that address the issues above, evaluating the feasibility of the solutions, and choosing the appropriate solution.

3. Carry out a detail-level design using appropriate design tools and methodologies.

4. Test and refine the implementation until the product or process design specifications are met or exceeded.

5. Document the finished product or process as appropriate for the discipline according to standard practice.

6. Present and transfer the product or process and documentation to the client.

In summary, the ILOs as they apply to the computer science program are:

| | |
|---|---|
| RH-1 | Ethics |
| RH-2 | Contemporary Issues |
| RH-3 | Global |
| RH-4 | Culture |
| RH-5 | Teams |
| RH-6 | Communication |
| RH-CSSE-7 | Problem Solving |
| RH-CSSE-8 | Experiments |
| RH-CSSE-9 | Design |

**Response to Institutional Concern from 2000 ABET EAC Final Report**

The 2000 ABET EAC Final Report indicated that a concern remained regarding the Rose-Hulman Mission Statement. The concern centered on the word “best” and the institution’s ability

to "operationalize" that term sufficiently to determine whether progress is being made in fulfilling that mission.

One could certainly substitute a less ambitious phrase for "world's best." One might use "excellence" as in the Polytechnic University, Drexel University, or University of Texas-Austin engineering college statements.

> ...to provide students with *an excellent* undergraduate education in engineering, mathematics, and science in an environment of individual attention and support.

But this statement, too, takes on meaning only when further defined through educational objectives and outcomes. Similarly, we argue that when viewed in concert with our list of six guiding principles, five institutional goals, 10 learning outcomes, and individual program educational objectives and outcomes, there is no ambiguity regarding what Rose-Hulman is all about.

Furthermore, we believe a proper mission statement serves a purpose over and above a simple statement of what an organization does. At Rose-Hulman, the mission statement serves as an inspirational statement that everyday reminds us that second best is not good enough. Many a decision concerning facilities, curriculum, and staffing are made easily once the question is asked of ourselves, "Is that the implementation that will help us become the best?"

Thus, we prefer a mission statement that is both descriptive and inspirational such as CWR's "Case Western Reserve University strives to be the most powerful learning environment in the world." Such mission statements declare not only what we do, but also the standards against which we will judge our progress.

We, therefore, request that ABET EAC and CAC accept our mission statement as created by faculty, staff, students, and the Board of Trustees.

### *Alignment of Program Educational Objectives with Mission Statement*

The mission statement is very broad in the sense that it is not specific to any particular career path or life situation. The ILOs are similarly broad. For example, it is difficult to imagine any societal role for which achieving the Teams and Communication ILOs would not be useful preparation. However, in some cases the relationships between the ILO's and PEO's are more direct than in other cases. Some of the more direct relationships are indicated in **Error! Reference source not found.**, which demonstrates that the computer science PEO's align well with the Rose-Hulman mission statement. The PEO's are discussed in detail in the following section.

**Table 1. Alignment of Program Educational Objectives with RHIT Mission Statement.**

| PEO | RH-1 Ethics | RH-2 Contemporary Issues | RH-3 Global | RH-4 Culture | RH-5 Teams | RH-6 Communication | RH-CSSE-7 Problem Solving | RH-CSSE-8 Experiments | RH-CSSE-9 Design |
|---|---|---|---|---|---|---|---|---|---|
| PEO-1 | X | X | | | X | X | X | X | |
| PEO-2 | X | X | X | X | X | X | | | |
| PEO-3 | X | X | | X | X | X | | | X |
| PEO-4 | | X | | X | X | X | | | |
| PEO-5 | X | X | X | X | X | X | | | |
| PEO-6 | | | | | X | X | | | |
| PEO-7 | X | X | | | X | X | X | | X |

***Program Educational Objectives***

The current computer science PEOs are

to prepare students to become

PEO-1 Graduate students and researchers.

PEO-2 Leaders in government and law as government employees, policy makers, governmental advisors, and legal professionals.

PEO-3 Entrepreneurial leaders.

PEO-4 Business leaders within existing organizations.

PEO-5 Actively involved in social and professional service locally, nationally, and globally.

PEO-6 Recognized by their peers and superiors for their communication, teamwork, and leadership skills.

PEO-7 Software professionals in a variety of organizations, including ones doing traditional software development, technological innovation, and cross-disciplinary work.

The PEOs are documented on the department web site, the URL for which is

http://www.cs.rose-hulman.edu/

The PEOs will be included in future Undergraduate Bulletins, but are not included in the AY2005-2006 bulletin, which was published before the PEOs were developed.

***Alignment of Program Educational Objectives with Needs of Constituencies***

The computer science program constituents (stakeholders) are:

- the program's current students,
- the program's alumni,
- the employers of the program's alumni, and
- the graduate programs to which the program's alumni apply.

***Assessment and Evaluation of Program Educational Objectives***

The assessment and evaluation of the computer science PEOs is accomplished as part of the department's overall process of continuous improvement, which is described in detail below. Achievement of each of the PEOs is measured by two mechanisms: an annual survey of the program's alumni and meetings with the CSSE Board of Advisors. Specifically:

- The survey of the program's alumni is administered annually by Rose-Hulman's Office of Institutional Research, Planning, and Assessment (IRPA). The surveys completed during AY 2005-2006 and earlier consisted entirely of questions that are provided by IRPA and are common to surveys of alumni from all Rose-Hulman programs. Future surveys will include questions provided by individual programs for their alumni. The program-specific questions for computer science program alumni will directly measure all of the PEOs, as well as direct feedback on the PEOs themselves.
- The CSSE Board of Advisors provides indirect measures of the achievement of all of the PEOs, and direct feedback on the PEOs themselves. The department reviews the PEOs with the Board of Advisors at least annually.

Upon receiving notification that the computer science program had been selected as a pilot program for the new CAC criteria, the department began the formal process of developing PEOs. The first step taken was asking the CSSE Board of Advisors for its view on what the program's alumni should be doing three to five years after graduation. The Board of Advisors is a valuable source of input from the three sets of constituents with the best perspective on what students can expect after graduation (alumni, employers, and graduate programs).

During the next department meeting, the department faculty addressed the same question. Over the course of the next several meetings, the faculty developed objectives by first merging and condensing these two independent sets of input, restating them in a measurable form where possible, and prioritizing them to reflect the essential aspects of the program. The PEOs will continue to evolve through the process described below, which incorporates student input as well as additional input from alumni and graduate programs.

At the time of this writing, a complete set of assessment data is available by which to evaluate the extent to which the PEOs are being met. That evaluation will be documented in the program assessment report, along with recommended changes to the program. The program assessment report and the status of the recommended changes will be available at the time of the accreditation visit.

***Program Outcomes***

The current computer science POs are that at the time of graduation

| | |
|---|---|
| graduates will demonstrate the abilities to: | |
| PO-1 | Apply effectively a variety of programming languages, programming paradigms, operating systems, networks, and software development tools. |
| PO-2 | Anticipate complexities and problems involved in the development of large software systems. |
| PO-3 | Analyze requirements, design software that satisfies those requirements, and implement that software. |
| PO-4 | Analyze problems using ideas of problem complexity, models of computation, and decidability. |
| PO-5 | Design algorithms using a variety of paradigms |
| PO-6 | Analyze algorithms in terms of correctness, as well as time and space efficiency. |
| PO-7 | Communicate effectively, both verbally and in writing. |
| PO-8 | Evaluate and discuss the legal, social, and ethical aspects of significant events that arise in the computing industry. |
| PO-9 | Collaborate effectively in small teams. |

The POs are documented on the department web site, the URL for which is

http://www.cs.rose-hulman.edu/

The POs will be included in future Undergraduate Bulletins, but are not included in the AY2005-2006 bulletin, which was published before the POs were developed.

***Assessment and Evaluation of Program Outcomes***

The assessment and evaluation of the computer science POs is accomplished as part of the department's overall process of continuous improvement, which is described in detail below. Achievement of the POs is measured by a combination of three mechanisms: course assessment reports, portfolio assessment, and senior exit interviews. Specifically:

- For every course offering, the instructor(s) write an assessment report that includes quantitative assessments of the course learning outcomes, a qualitative assessment by the instructor(s), and a summary of the student feedback. The reports are reviewed and approved by the department faculty, ideally at the department meeting following the deadline for course grades. Parts of these reports provide direct measures of the achievements of all of the POs. The course assessment process is described in detail below.
- Selected student work is submitted to the RosE Portfolio System (REPS) to assess each student's achievement of the ILOs. On an annual basis, Rose-Hulman faculty rate the submissions, and IRPA provides a report assessing the performance of each program's students. This report provides direct measures of the achievement of POs 7-9. REPS is described in detail below.
- Each computer science student who is eligible to participate in the annual commencement exercises near the end of May is invited to participate in a one-on-one exit interview with the Head of Computer Science and Software Engineering. As part of the interview, the students are asked to rate how well they feel they have satisfied each of the computer science POs.

The specific mechanisms that are used to assess and evaluate each of the PO's are indicated in Table 2.

**Table 2. Mechanisms used in the assessment and evaluation of program outcomes.**

| PO | Course Assessment Reports | Portfolio Assessment | | | | | | | | | Exit Interviews |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RH-1 Ethics | RH-2 Contemporary Issues | RH-3 Global | RH-4 Culture | RH-5 Teams | RH-6 Communication | RH-CSSE-7 Problem Solving | RH-CSSE-8 Experiments | RH-CSSE-9 Design | |
| PO-1 | X | | | | | | | | | | X |
| PO-2 | X | | | | | | | | | | X |
| PO-3 | X | | | | | | | | | | X |
| PO-4 | X | | | | | | | | | | X |
| PO-5 | X | | | | | | | | | | X |
| PO-6 | X | | | | | | | | | | X |
| PO-7 | X | | | | | | X | | | | X |
| PO-8 | X | X | X | | X | | | | | | X |
| PO-9 | X | | | | | X | | | | | X |

In the fall of AY 2005-2006, the department faculty began the formal process of developing POs by identifying outcome statements in the catalog description of the program. The faculty also asked the CSSE Board of Advisors for its view on what the program's alumni should be able to do at the time of graduation. These inputs were combined and refined in a series of faculty meetings, resulting in the POs listed above. The POs will continue to evolve through the department's continuous improvement process, which is described in the following section.

At the time of this writing, a complete set of assessment data is available by which to evaluate the extent to which the PEOs are being met. That evaluation will be documented in the program assessment report, along with recommended changes to the program. The program assessment report and the status of the recommended changes will be available at the time of the accreditation visit.

### *Alignment of Program Outcomes and Program Educational Objectives*

Each of the POs contribute directly to the preparation of graduates to exhibit the qualities required by at least some of the PEOs. The converse is also true – each of the PEOs is supported by at least some of the POs. Some of the most direct relationships are indicated in Table 3.

**Table 3. Alignment of Program Outcomes and Program Educational Objectives.**

| PEO | PO-1 | PO-2 | PO-3 | PO-4 | PO-5 | PO-6 | PO-7 | PO-8 | PO-9 |
|---|---|---|---|---|---|---|---|---|---|
| PEO-1 | X | X | X | X | X | X | X | X | X |
| PEO-2 | | X | | | | | X | X | X |
| PEO-3 | X | X | X | X | X | X | X | X | X |
| PEO-4 | | X | | | | | X | X | X |
| PEO-5 | | | | X | | X | X | X | X |
| PEO-6 | | | | | | | X | | X |
| PEO-7 | X | X | X | X | X | X | X | X | X |

## B. Student Attributes:

The purpose of this section is to address the following question:

> Does the program enable students to achieve the attributes listed above by the time of graduation?

The department is confident that the computer science program's students achieve the ABET CAC attributes, including those for CS programs. With the exception of ABET CAC attribute (h) above, this achievement is demonstrated by the student's achievement of the POs and ILOs, as illustrated in Table 4. There are no ABET CAC attributes that are assessed only by the Design ILO.

Direct evidence that the program's students achieve ABET CAC attribute (h) includes the course assessment report for CSSE 372. Indirect evidence includes the results of the software engineering exit interviews, in which 100% of software engineering graduates indicated satisfaction that "by graduation, students in the software engineering program will have the knowledge required to understand the need for and the ability to perform in lifelong learning." The courses that contribute most directly to this outcome of the software engineering program are CSSE 120, 220, 230, 332, and 372. All of these courses are also required for the computer science program. Furthermore, the need for lifelong learning is a strong emphasis of CSSE 304, which is required for the computer science program.

In order to ensure that the computer science program's students continue to achieve ABET CAC attribute (h) in the future, the department will add a PO related to lifelong learning.[7] Regardless, a question about lifelong learning will be added to the computer science program's exit interview to provide a second assessment tool for this attribute.

[7] This will require a vote of the department faculty, which will take place prior to the ABET visit in AY 2006-07. It is difficult to imagine a dissenting vote.

**Table 4. Alignment of CAC attributes with Program Outcomes and Institutional Learning Outcomes.**

| | POs | | | | | | | | | ILOs | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Attributes | PO-1 | PO-2 | PO-3 | PO-4 | PO-5 | PO-6 | PO-7 | PO-8 | PO-9 | RH-1 Ethics | RH-2 Contemporary Issues | RH-3 Global | RH-4 Culture | RH-5 Teams | RH-6 Communication | RH-CSSE-7 Problem Solving | RH-CSSE-8 Experiments | RH-CSSE-9 Design |
| CAC-CS-a | X | X | X | X | X | X | | | | | | | | | | X | | X |
| CAC-CS-b | | | X | X | | | | | | | | | | | | X | | |
| CAC-CS-c | X | X | X | | X | X | | | | | | | | | | | | X |
| CAC-CS-d | | | | | | | | | X | | | | | X | | | | |
| CAC-CS-e | | | | | | | | X | | X | X | X | X | | | | | |
| CAC-CS-f | | | | | | | X | | | | | | | | X | | | |
| CAC-CS-g | | | | | | | | X | | | | X | | | | | | |
| CAC-CS-h | | | | | | | | | | | | | | | | | | |
| CAC-CS-i | X | | | | | | | | | | | | | | | X | | |
| CAC-CS-j | | | | X | X | X | | | | | | | | | | X | | |
| CAC-CS-k | | X | | X | X | | | | | | | | | | | | | X |

## C. Assessments and Program Improvement

The purpose of this section is to address the following question:

> Does the program use a process incorporating relevant data to regularly assess its educational objectives and outcomes, and to evaluate the extent to which they are being met? Are the results of the evaluations used to develop and implement plans to effect continuous improvement of the program?

As such, it presents and discusses the following:

- the department's continuous improvement process; and
- past, ongoing, and planned program improvements.

### Continuous Improvement Process

The department has implemented a Continuous Improvement Process (CIP) that serves to assess and evaluate both the computer science and software engineering programs. The computer science Program Assessment Plan (PAP), which is developed by the Computer Science Program Coordinator and approved by the department faculty prior to the start of each academic year, specifies the mechanisms by which each of the POs and PEOs is assessed and evaluated. The data collected are incorporated in the computer science Program Assessment Report (PAR), which is compiled by the program coordinator at the end of each academic year. Based on the assessment data, the PAR also evaluates the extent to which each PO and PEO is being achieved. The department faculty then use the PAR to develop plans by which to make improvements to the program, which are discussed in department and committee meetings and recorded in the minutes of those meetings.

As stated above and illustrated in the figure below, the mechanisms that are used to assess and evaluate the achievement of the POs are course assessment, portfolio assessment, and senior exit interviews. For the sake of clarity, input from sources other than course assessment is represented in summary form in the figure.

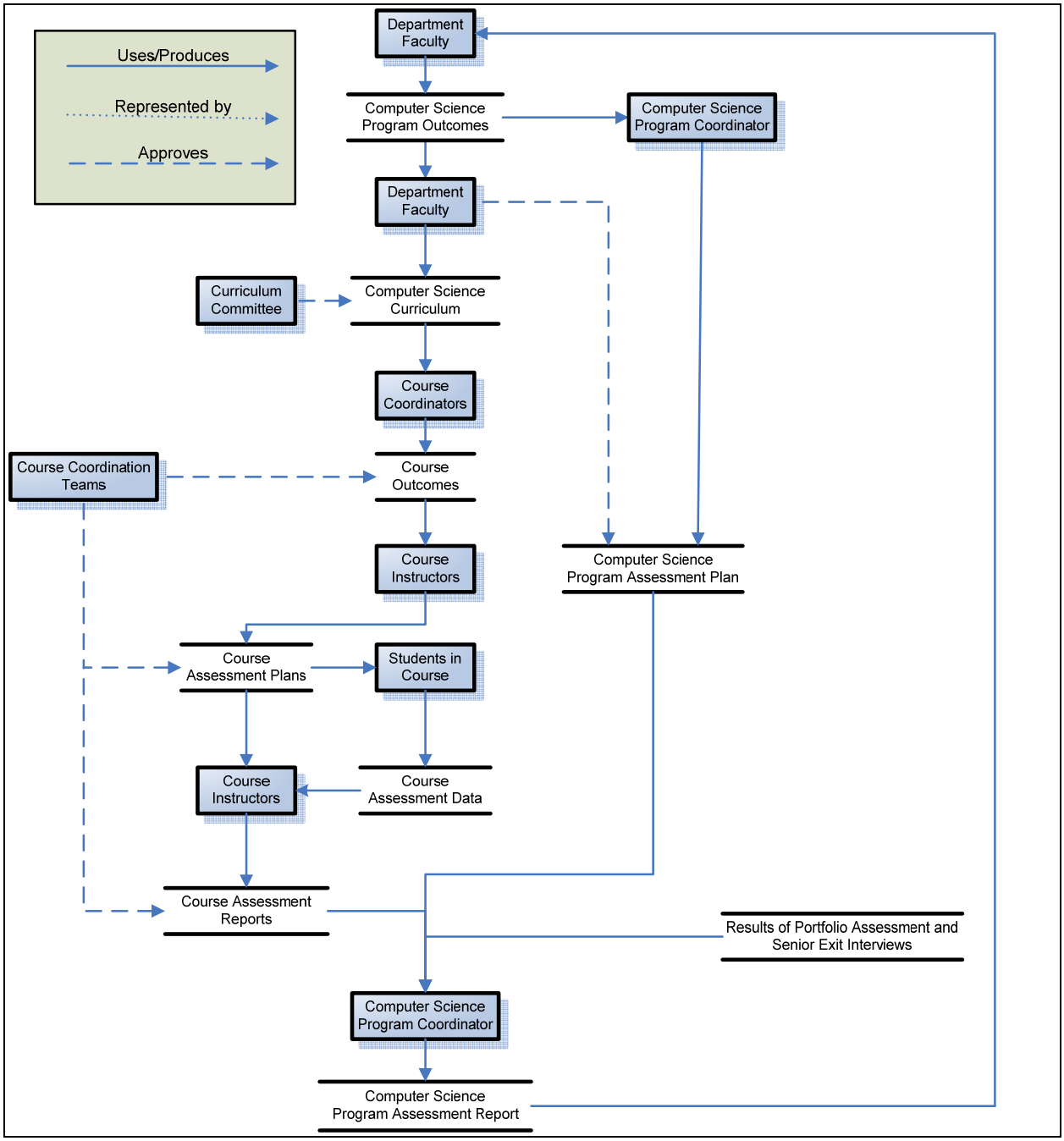



**Figure 1. Elements of CSSE Continuous Improvement Process that are relevant to Computer Science Program Outcomes.**

Similarly, the mechanisms that are used to assess and evaluate the achievement of the PEOs are an alumni survey and meetings with the CSSE Board of Advisors, as stated above and illustrated in the figure below.

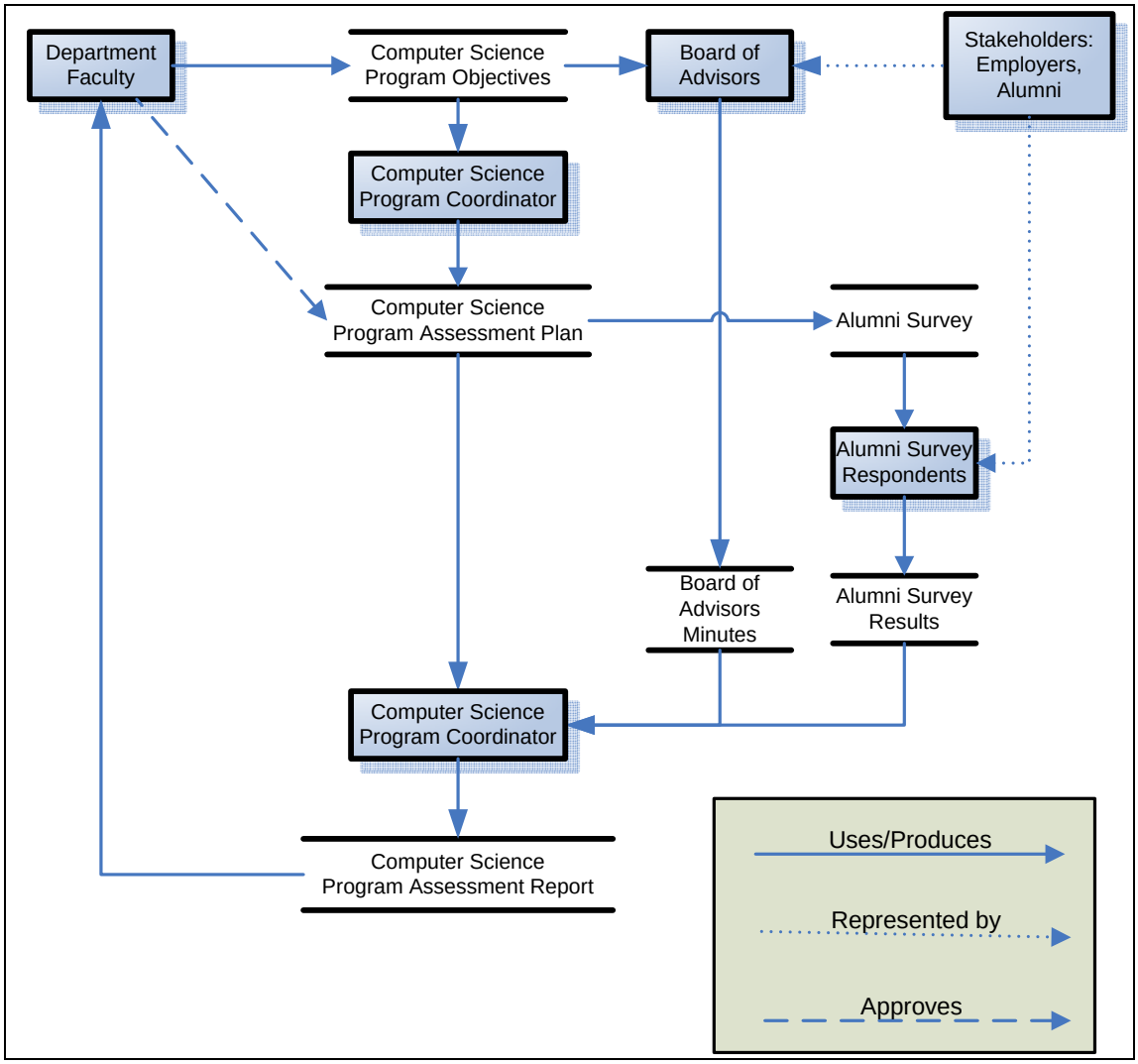



**Figure 2. Elements of CSSE Continuous Improvement Process that are relevant to Computer Science Program Educational Objectives.**

The remainder of this section discusses the aspects of the CIP that are most relevant to the CAC accreditation criteria. Specifically, it addresses:

- frequency and timing of assessments,
- what data are collected,
- how data are collected,
- from whom data are collected, and
- how assessment results are used and by whom.

It does this by first discussing each of the assessment mechanisms in detail, and then summarizing the information relevant to the items listed above.

***Assessment Mechanisms***

The CIP uses five assessment mechanism, which this section discusses in turn:

- Course assessment
- Portfolio assessment
- Senior exit interviews
- Alumni surveys
- CSSE Board of Advisors

**Course Assessment**

Each course has a Course Coordination Team (CCT). Each team has three long term members, one being designated the Course Coordinator. In addition, recent, current, and projected instructors of the course are de facto members of the team. The long term members of the team are chosen with the intent of incorporating as many diverse perspectives as possible into the assessment and evaluation of the course. Specifically, the coordinator is a department faculty member who has taught the course or is likely to teach it in the future. One of the other members of the team is one of the two program coordinators, usually the one from the program that is less directly impacted by the course (if such a distinction can be made). The third member is a department faculty member who has not had much other direct involvement with the course, and who is not likely to in the future. This membership scheme has the additional benefit that each faculty member maintains a certain level of familiarity with courses throughout the department, not just those that they tend to teach frequently.

The CCT is responsible for producing the documents related to the assessment of that course, which consist of the Course Outcomes (COs), the Course Assessment Plan (CAP), and the Course Assessment Report (CAR). The documents are then reviewed and approved by the department faculty as a whole.

The CO document contains only the learning outcomes for the course, which define the abilities that students should have upon successfully completing the course. The department faculty have committed in principle to having the students' achievement of the COs as the primary goal of the courses they teach, although each instructor remains free to have additional course goals. As such, the COs represent a fundamental mechanism by which to ensure the stability of the program, as well as to implement needed changes, thereby making the CIP possible. Given the fundamental nature of the COs, the department has invested considerable effort in developing them in such a way that they are complete, specific with respect to the level of learning intended, and yet not prescriptive in terms of course implementation.

The CAP for a course offering describes how each CO will be assessed quantitatively (and restates the COs so that the document is self-contained). In essentially every instance, each CO is assessed using two independent tools. The CAP also defines the success criteria for each assessment of each CO. The assessment tools typically consist of the relevant portions of some combination of graded work, such as homework, quizzes, exams, and project reports, but instructors are free to use any assessment tool. As such, the CAPs tend to change more frequently than the COs, and in particular based on which instructor(s) are teaching the course.

The CAR for a course offering is the long term record of all assessment related information for that offering, and as such includes all of the information from the CAP for that offering. It also

summarizes the results of the quantitative assessments specified in the CAP for that offering. Furthermore, it includes[8]

- the recommendations from the previous offering,
- how those recommendations were addressed,
- other changes from the previous offering,
- qualitative observations by the instructors
- a summary of relevant student feedback
- recommendations for the next offering

Because each CAR includes the recommendations from the previous offering and how they were addressed, as well as recommendations for the next offering, the course assessment process forms the inner loop of the CIP. The idealized timeline for each course offering is as follows:

- CCT reviews CAR from previous offering, including recommendations for this offering
- CCT proposes necessary changes to COs
- Department approves changes to COs
- Instructors plan modifications to course implementation
- CCT proposes necessary and desired changes to CAP
- Department approves changes to CAP
- Instructors implement modifications to course implementation
- Instructors teach course and collect assessment data
- CCT proposes CAR, including how recommendations from previous offering were addressed and recommendations for next offering

This timeline was not always followed during AY 2005-2006, but it should happen routinely in the future now that the teams have existing documents to work from and the faculty are familiar with the process.

**Portfolio Assessment**

### RHIT Portfolio Assessment Process

The RosE Portfolio is an online, digital system developed at Rose-Hulman Institute of Technology. It is designed to allow students, faculty, and administrators to archive, assess, and evaluate student work for the purpose of class, department, program, and institutional assessment.

[8] The descriptions of the course assessment documents given here represent the minimum amount of information that the department faculty have committed to including. Individual instructors are also free to include additional information in any of the documents. In particular, instructors are free to include program recommendations, which are then incorporated into the program assessment report and become part of the continuous improvement process at the program level.

In addition, the RosE Portfolio offers features like the Showcase Resume, the Curriculum Map, and customizable Assessment Outlines.

The Commission on the Assessment of Student Outcomes (CASO) began development of the RosE Portfolio in 1996. CASO established two objectives in developing an electronic portfolio for the purpose of student learning outcomes assessment. First, CASO wished to create a data collection method that would allow students to submit documents they produced in technical and non-technical courses (authentic evidence), rather than using standardized tests, documents produced solely for the purposes of assessment, or surveys. Second, CASO wished to develop an online portfolio that would allow for submission, rating, and evaluation of evidence of student learning within a single electronic system. While the RosE Portfolio System (REPS) has been used since 1998, a new REPS software product was developed in the summer of 2004 in cooperation with Quest Information Systems, Inc. The 2005 Rating Session was the first time the new software was used for a full rating session.

REPS consists of the following components:

- Student submission interface: allows students to submit documents in any format to an established set of student learning outcomes
- Faculty interface: allows faculty to verify that students have made the required submissions to the appropriate student learning outcomes; also allows faculty to use the Curriculum Map function, a process in which the faculty member maps his/her course outcomes to Institute-level outcomes
- Rater interface: allows faculty raters to assess student submissions against a defined set of evaluation rubrics. The results of the rating sessions are reported back to all departments on campus and are applied to the further development of the programs.

### *Purpose of Rating*

The purpose of the RosE Portfolio (REPS) Rating Session is to assess evidence of student learning in the six non-technical ILOs. Each ILO is defined by a set of performance criteria, as described above.

Evidence of student learning in these six ILOs is collected each year through assignments made by faculty in technical and non-technical departments. For example, some engineering faculty require that students submit documents from capstone senior design courses as evidence for the Teams ILO. Furthermore, Humanities and Social Sciences faculty require that students submit documents produced in their courses for evidence of the Global and Culture ILOs. Although evidence of student learning in all six ILOs is collected every year, the student submissions to the portfolio are rated on a regular cycle, as shown in Table 5.

**Table 5. REPS rating cycle.**

| ILO | 2003-04 | 2004-05 | 2005-06 | 2006-07 | 2007-08 | 2008-09 |
|---|---|---|---|---|---|---|
| RH-1 Communications | ✓ | | ✓ | | ✓ | |
| RH-2 Ethics | ✓ | | ✓ | | ✓ | |
| RH-3 Teams | ✓ | | ✓ | | ✓ | |
| RH-4 Contemporary Issues | | ✓ | | ✓ | | ✓ |
| RH-5 Culture | | ✓ | | ✓ | | ✓ |
| RH-6 Global | | ✓ | | ✓ | | ✓ |

Definition, collection of documents, and assessment and evaluation of evidence for the remaining (technical) ILOs, both at the Institute and department levels, is the province of technical departments. Some departments elect to use REPS for this data collection and assessment. The specific procedures used by CSSE are described below.

### *Rating Scope and Methodology*

Rating submissions to the RosE Portfolio has followed the same methodology since the system was initiated in 1998. Rose-Hulman faculty are hired as portfolio raters. Attempts are made to involve faculty from many different departments on campus to ensure objectivity in rating and broad based familiarity with and participation in the process. Raters work for two days in the Dynamics Laboratory (Olin 203) and are compensated at the rate of $250 per day. The Rating Session Coordinator facilitates the process and assigns pairs of raters to rate student submissions for a particular ILO. For example, a mechanical engineering faculty member and a chemistry faculty member may work as a rating pair assessing the student files under Criterion 4 of the Communication ILO.

The rating process consists of three steps.

1. First, REPS requires that the faculty rater pair rate a set of three shared documents. The rating is made on the basis of a Rating Rubric and uses a Yes/No benchmark for a single rating question: "Does this submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman." The raters must agree in their rating of the initial document set. This step is called Inter-Rater Reliability (IRR). The process ensures that raters look for the same qualities and components in order to rate documents as acceptable.
2. Second, if the raters agree in their IRR, the system then allows them to proceed with a set of ten documents, each rater reading and rating a different set of ten. The system records their rating for each document. The system also introduces a shared file every twelve documents in order to check that the raters have maintained their Inter-Rater Reliability. Failure to rate the shared document identically will cause the system to stop the raters so that they can recalibrate their work before moving on to another document set, ensuring validation of the rating process.
3. Third, the raters can provide comments about the rating session or about the student submission in the Comment boxes. In addition to the work of rating, faculty raters are asked to identify Exemplary Documents among the files that they assess.

For the Rating Session, portfolio raters use the ILO, Performance Criteria, and Rubrics that have been developed by CASO based on feedback from faculty in all departments. The list of faculty raters, their home departments, and their assigned performance criteria is provided in Table 6 (from the most recent Rating Session, Summer 2005).

**Table 6. Faculty raters for Summer 2005 Rating Session.**

| ILO | Criterion | Rater | Department |
|---|---|---|---|
| Ethics | 1 and 2 | Heinz Luegenbiehl | HSS |
| | 1 and 2 | Anneliese Watt* | HSS |
| Communication | 1 and 3 | Rebecca DeVasher | Chemistry |
| | 1 and 3 | Bill Weiner * | ABBE |
| | 2 | Glen Livesay | ABBE |
| | 2 | Don Richards* | ME |
| | 4 | Bruce Black | ECE |
| | 4 | Clark Merkel | ME |
| | 4 | Michael Mueller* | Chemistry |
| | 4 | Renee Rogge | ABBE |
| Teams | 1 | Mark Ardis* | CSSE |
| | 1 | Jim Hanson | CE |
| | 1 | Sudipa Kirtley* | PHOE |
| | 1 | Bill Kline | EM |
| | 2 and 3 | David Erwin* | Chemistry |
| | 2 and 3 | Richard House | HSS |

Faculty who have served as raters in previous Rating Sessions are marked with an "*".

Two key components characterize REPS and distinguish it from other data assessment methods. First, faculty raters are not reading and evaluating documents that were produced in their own courses. Instead faculty rate submissions independent from the courses in which they were produced. This contributes to the validity of the rating process. Second, faculty evaluate documents against a set of defined rubrics (as described above).

After the Rating Session results are compiled, they are used by technical departments for preparation of their own ABET reports, and by the academic dean and technical departments for their Higher Learning Commission accreditation reports, as well as reporting to constituencies interested in the evaluation and continuous improvement of RHIT.

### *Rating Rubrics*

The rubrics against which the student work is rated for each of the performance criteria of the Ethics, Contemporary Issues, Global, Culture, and Teams, and Communication ILOs are given below. As previously stated, the performance criteria (and rubrics) for the remaining ILOs are determined independently by each department. The criteria and rubrics used by CSSE are given in the next section.

RH-7 Ethics - A recognition of ethical and professional responsibilities. When given the opportunity, students will:

1. Demonstrate knowledge of a professional code of ethics.

- Description: The student should clearly establish that he/she has read and understands a professional code of ethics (e.g., by quoting a specific passage, referencing a specific verse, or otherwise displaying an intimate knowledge of the code).
- Example: A passage or section from the code of an engineering professional organization (e.g., AICHE, NSPE, ASCE, etc.), could be cited, or one from another professional organization code (e.g., the American Psychological Association code could be cited by students in a psychology class).
- Not Acceptable: Citing the RHIT student code of ethics, or other unrelated codes of conduct, like the Constitution, the Ten Commandments, or engineering codes and standards that deal with issues like details of construction, testing, or product specifications (e.g., ASME Pressure Vessel Code, ASTM specifications, building codes, or safety codes).

2. Evaluate the ethical dimensions of professional engineering, mathematical, and scientific practices.

- Description: The student should describe an application of a professional code of ethics related to engineering, mathematics, or science, with a clear connection between the code provision and the application.
- Example: A case study, a personal experience, or a debate about a contemporary issue could be used as long as there is an ethical dimension involved.
- Not Acceptable: Describing an ethical issue outside of engineering, mathematics, and science (such as social science) or an application that does not clearly involve ethics.

RH-8 Contemporary Issues - An understanding of how contemporary issues shape and are shaped by mathematics, science, & engineering When applying the principles of mathematics, science, and/or engineering to a technical problem, students will:

1. Demonstrate an awareness of how the problem is affected by social concerns and trends.

- Description: The student should define the technical problem and demonstrate the link between it and social concerns/trends. A student should recognize the larger social context in which the problem emerges.
- Example: A student may link the oil crisis to the development of hybrid cars.
- Not Acceptable: A student describes a typical day in his/her life, there is no discussion of the larger social context, or the problem is not a technical problem.

2. Demonstrate an awareness of how the proposed solution(s) will affect culture and the environment.

- Description: The student should offer a proposal for solving the problem posed in
- Criterion A. The link between the solution and the culture at large/environment must be explicit.

- Example: The student could evaluate the hybrid car solution to the oil crisis by measuring pollution levels.
- Not Acceptable: The cultural/environmental effects are not evaluated, or the problem is not a technical problem.

RH-9 Global - An ability to recognize the impact of global societies on citizens and professionals. When given the opportunity, students will:

1. Demonstrate an awareness of the historical development of cultures and societies.
   - Description: The student discusses/summarizes the societal impact of events, processes, and/or developments. The student may deal with only one nation or culture, or with several.
   - Example: A student could examine any of the following: cases of cause and effect resulting from an historical event; the history of a culture within a county; historical events (political alliances, treaties, wars, social attitudes) with an emphasis on historical development.
   - Not Acceptable: The student provides only a chronological statement of events, or no evidence is provided. The student does not provide citation to appropriate references to support the thesis of the submission.
2. Show an awareness of the relationships of nations and the interdependence of peoples around the globe.
   - Description: The student demonstrates awareness of how two cultures or nations change and influence each other.
   - Example: A student could submit a paper on one of the following: an analysis of how the actions of one or more countries affect each other; a discussion of international relations; a paper from a course in sociology, anthropology, geography, political science, economics, or history when the analysis deals with more than one nation.
   - Not Acceptable: The submission only deals with a single people or culture, or there is no evidence of awareness of the relationships and/or interdependence. The submission only provides a comparison of two forms of government without analysis of their affect on one another; papers that simply contrast two forms of government but do not show the interdependence of the two forms.

RH-10 Culture - An ability to understand diverse cultural and humanistic traditions. When given the opportunity, students will:

1. Perform, interpret, analyze, or otherwise engage in artistic, literary, and/or other forms of culture.
   - Description: The student demonstrates exposure to or engagement in artistic, literary, and/or other forms of culture.
   - Example: The student should document a non-technical, creative, or artistic expression of any culture: any assignment that comes from a literature class; analysis of a work of literature; a review of a student's work in a drama production or musical performance.

- Not acceptable: There is no evidence of exposure to or engagement in artistic, literary, and/or other forms of culture. Translations of a document into another language; ethics papers, technical communication papers, social issues papers written for RH 131.

2. Interpret, analyze, or evaluate aspects of a society other than the student's own.

   - Description: The student analyzes characteristics of a society or culture other than his/her own, discusses the importance of contributions of people from other cultures, or evaluates an issue or problem from a cultural perspective other than his/her own.

   - Examples: The student could submit any of the following: a journal entry from a trip to a foreign country; a paper on a technology whose development was influenced by another culture(s); a paper that includes an analysis of a minority culture within the United States; a paper that examines a polticial issue from the perspective of a non-U.S. country; a paper that examines religion from a non-U.S. perspective.

   - Not acceptable: A paper in which there is no interpretation, analysis, or evaluation of a society or culture other than the student's own. A paper in which a student dismisses cultures and/or societies other than his own.

RH-11 Teams - An ability to work effectively in teams. When assigned to teams, students will:

1. Share responsibilities and duties.

   - Description: The student should describe the roles of each team member and indicate how sharing responsibilities facilitates project completion.

   - Example: In a team member evaluation, a student might list the tasks performed by each member of the group. A student might be team leader for one meeting, recorder for another meeting, responsible for generating AutoCad drawings, and check design calculations.

   - Not Acceptable: To list only the student's own responsibilities and only one role performed.

2. Analyze ideas objectively to discern feasible solutions and then build consensus.

   - Description: The student should indicate which ideas were considered and how the ideas were evaluated, discuss the outcome of the analysis of ideas with a brief justification, and articulate the process the group used for determining their final solution.

   - Example: A student might list pros and cons for multiple ideas, might justify the group's decision based on economic factors, and might submit minutes from the group meeting where the final solution was selected.

   - Not Acceptable: A list of ideas without analysis, selecting an idea without correlating it to some selection criteria, and submitting the final report without explanation of the process by which consensus was reached.

3. Develop a strategy for action.

   - Description: The student should identify the tasks necessary for successful completion of the group project.

- Example: A student might use a Gantt chart to prepare a timetable for completion.
- Not Acceptable: Failure to identify either tasks or milestones for completion.

RH-12 Communication - An ability to communicate effectively in oral, written, graphical, and visual forms. When performing communication tasks, students will:

1. Identify readers/audience, assess their previous knowledge & information needs, and organize/design information to meet the needs.
   - Description: The student should give a brief description of the readers/audience for the document or project, an assessment of their needs, and a demonstration of how the information design accommodates that audience.
   - Example: A student may include an audience analysis worksheet or a usability study of the audience. A student could draw the rater's attention to a glossary of terms included in the document in order to accommodate readers less familiar with the topic.
   - Not Acceptable: A statement of the intended audience without an analysis of their previous knowledge or information needs, or a statement that the student accommodated readers without reference to particular features of the document or project.
2. Provide content that is factually correct, supported with evidence, explained with sufficient detail, and properly documented.
   - Description: The student should verify the information provided with reference to sources/authorities and document those authorities in a style appropriate to his/her discipline (IEEE, APA, MLA). The student should also provide adequate detail to assist the readers' understanding of the topic.
   - Example: A student could submit a Technical Communication report that documents statements cited from authoritative sources and provides a bibliography of such sources.
   - Not Acceptable: Any document that offers inadequate information without attribution or any document that appears to be plagiarized from an authoritative source.
3. Test readers/audience response to determine how well ideas have been relayed.
   - Description: The student should discuss how she tested her document/project on readers/audience and how that testing determined revisions to the document.
   - Example: A student may submit the results of a usability study that demonstrates how effectively the document met her readers' needs for information and discuss what changes were made to the document based on the results of the study.
   - Not Acceptable: No evidence of reader/audience testing of the document/project.
4. Submit work with a minimum of errors in spelling, punctuation, grammar, and usage.

- Description: The student should submit a document/project that has a low percentage of mistakes.
- Example: A student may submit a final design report that has been carefully proofread and edited.
- Not Acceptable: A level of errors that is distracting or noticeable, such as finding errors in a one or two page sample of text, or in a conspicuous place such as a title, a heading, or an important term.

### *Portfolio Usage Comparison*

In the 2003 Rating Session, 2,207 unique submissions were made to REPS. During the 2005 Rating Session, 6,928 unique submissions were made to REPS. This represents a significant increase in usage of the system since 2003, the last time that submissions to the Communication, Ethics, and Teams ILOs were rated. In 2003, only documents collected during one academic year were rated. In 2005, we rated documents that had been collected for two years, reflecting the Portfolio Rating Cycle (see Table 1).

### *Global and Culture ILOs*

The Department of Humanities and Social Sciences supports the efforts of the Institute and all programs to document student learning in the Global and Culture ILOs. Each faculty member in the HSS Department identifies at least one course during the academic year in which students will make submissions to either the Global or Culture ILO. Frequently faculty members require that students make submissions in more than one of their courses. Then the faculty member designates a particular assignment that provides students with the opportunity to demonstrate their learning in the ILO. For example, a faculty member teaching in history requires all students in his class to submit the research essays they wrote examining the impact of a particular catastrophic event (like the devastating earthquake in Chile in the late 19th century) on social and political structures. The choice of the assignment only comes after the faculty member has considered the ILO and the performance criteria specified for it. Faculty are also encouraged to attend the Forums (offered three times per year) during which staff of the Office of Institutional Research, Planning and Assessment and faculty who have experience in assignment design provide assignment samples and discuss best practices.

The HSS faculty monitor student submissions to ensure that all students enrolled in their courses make the submissions. After the academic year is over, the submissions made by students in HSS courses are rated along with the rest of the submissions in the RosE Portfolio system. The files are rated by the cross-disciplinary team of faculty raters assembled for the Rating Session, and thus the files submitted in HSS courses are rated by faculty outside of the HSS Department.

Support of the Global and Culture outcomes assessment process has been growing within the HSS Department since the inception of the RosE Portfolio System. Rating results from the 2004 Rating Session (the last time the Global and Culture outcomes submissions were rated) reflected that not enough students were submitting documents to REPS. HSS faculty reviewed the submission process and determined that the Global and Culture outcomes and performance criteria were not worded in ways that accurately reflected the content and topics of the courses in the HSS Department. In the summer of 2004, immediately following the Rating Session, members of the department met to review and revise the outcomes and their performance criteria. These revisions were approved by the HSS Department, thus ensuring greater buy-in by the faculty. At that point, because HSS faculty had been involved in the revision of the performance criteria, they agreed to take a more active role in designing appropriate assignments and guaranteeing that students made submissions. The 2006 Rating Session should demonstrate that

the process has been improved as a result. In addition, the previous schedule of rating (only rating 3 outcomes per year on an alternating 3/3 system) has been replaced this year (2006) by an agreement to rate all files in all outcomes every year for the next 3 years. We believe this will help us achieve steady state both in student submissions to REPS and in rating results. By achieving steady state, we can make effective decisions regarding the future of REPS.

## CSSE Portfolio Assessment Process

As do all RHIT departments for all programs, CSSE uses the REPS Rating Process to assess the extent to which the computer science program achieves the Ethics, Contemporary Issues, Global, Culture, Teams, and Communication ILOs. The performance criteria that CSSE has defined for the Problem Solving, Experiments, and Design ILOs are given below, together with the methods used for assessing the extent to which computer science students achieve those criteria.

For the Problem Solving ILO, the department annually rates student work from CSSE220 and CSSE230 (Fundamentals of Software Development II and III against the rubrics shown for each performance criterion. Both courses are required for all computer science majors[9].

RH-CSSE-7 Problem Solving

1. Given a specific implementation of a programming problem, analyze the run time of that implementation.
    - Suggested artifact for assessment: CSSE230 (Fundamentals of Software Development III) Exam question
    - Description: Students should be able to establish that they understand how to find the runtime of code containing loops.
    - Example: When asked to find the complexity of a block of code including multiple loops, students should recognize that nested loops increase the complexity of code, while sequential loops do not. As a result, they will give the correct big-oh analysis in each case.
    - Not acceptable: Students give the wrong big-oh analysis for the nested loops or multiply individual loop runtimes when finding the runtime of a series of sequential loops.
2. Design an algorithm to solve a given problem efficiently.
    - Suggested artifact for assessment: CSSE230 (Fundamentals of Software Development III) Exam question
    - Description: Students should be able to produce an algorithm that correctly solves a given problem efficiently using the design conventions specified in the course.
    - Example: Given two sets implemented as sorted arrays, the student can correctly write an algorithm to find the union of the two sets. Key ideas

[9] Some students receive credit for these courses via advanced placement or transfer credit.

include incrementing the loop indices correctly to include all elements but avoid duplicates. The runtime should be linear in the size of the arrays.

- Not acceptable: Not enough detail is provided about when to increment the loop indices. All the elements in one array that are greater than the largest element in the other array are ignored because the student terminates the loop improperly. Solution ignores the fact that the arrays are sorted, thus requiring a quadratic algorithm.

3. Given a detailed specification of a programming problem, implement that specification in an appropriate programming language correctly.

   - Suggested artifact for assessment: CSSE220 (Fundamentals of Software Development II) BigRational lab exercise
   - Description: Students should implement the abs, add, divide, multiply, negate, and subtract methods along with appropriate constructors. The implementations should be correct and follow the style conventions specified in the course.
   - Example: See the sample solution to the BigRational project.
   - Not acceptable: One or more of the implementations are seriously incorrect. Style has gross violations of the style conventions specified in the course.

4. Given a proposed implementation of a detailed specification of a programming problem, unit-test that proposed solution in an effective manner.

   - Suggested artifact for assessment: CSSE220 (Fundamentals of Software Development II) BigRational lab exercise
   - Description: Students should provide unit-tests of the abs, add, divide, multiply, negate, and subtract methods. The unit-tests should be provided by using a tool like JUnit. The unit-tests need not be exhaustive but, taken collectively, should demonstrate an understanding of covering a variety of cases including extreme cases.
   - Example: See the sample solution to the BigRational project.
   - Not acceptable: One or more of the methods/constructors is not tested at all. (But note that it is normal for a JUnit test of one method to be an effective test of several methods.) The unit tests fail to include any tests of extreme cases. The unit tests fail to include more than one test per method/constructor.

For the Problem Solving ILO, the Department of Physics and Optical Engineering rates student work from PH 111 and 112 (Physics I and II) against the performance criteria shown. Both courses are required for all computer science majors[10]..

RH-CSSE-8 Experiments Criteria

[10] Some students receive credit for these courses via advanced placement or transfer credit.

1. Use appropriate experimental techniques and procedures
    - Provided by PH 111/112
2. Keep detailed and accurate records of experimental procedures and data collected
    - Provided by PH 111/112
3. Estimate experimental uncertainties and use appropriate error propagation techniques
    - Provided by PH 111/112
4. Draw reasonable conclusions from results
    - Provided by PH 111/112

For the Design ILO, the department rates student work from CSSE 497-498-499 (Senior Projects I-III) against the rubrics shown for each performance criterion. Both courses are required for all computer science majors except those whose senior thesis proposals are approved by the department faculty. Insufficient achievement of the Design ILO in previous work is one possible reason that a thesis proposal might be disapproved.

RH-CSSE-9 Design Criteria. When given the opportunity, students will:

1. Develop a design specification that addresses customer/client needs and constraints.
    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Requirements Specification
    - Description: The student should describe all of the customer's requirements for the product or process.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Requirements Specification including use cases that describe the flow of activities for the product or process.
    - Not acceptable: insufficient coverage of use cases
2. Carry out a conceptual design by generating multiple solutions that address the issues above, evaluating the feasibility of the solutions, and choosing the appropriate solution.
    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Architecture Document
    - Description: The student should create a high-level design whose features and capacities are traceable to the key requirements described in the problem statement and requirements document.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Architecture Document
    - Not acceptable: The architecture document does not describe how key required features and capacities will be achieved by that design.
3. Carry out a detail-level design and implementation using appropriate design tools and methodologies.
    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Software Deliverables

- Description: The student should detail their design in a manner that meets strategic goals of the project, such as implementing required features in a manner which easily can be maintained by others.
- Example: CSSE 497-498-499 (Senior Projects I-III) Software Deliverables
- Not acceptable: The implementation does not follow design guidelines expressed in the architecture document. For instance, a design template for error handling, required by the architecture, is not adhered to.

4. Test and refine the implementation until the product or process design specifications are met or exceeded.
   - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Test Results Report
   - Description: The student should describe the planned test cases and the results of running the tests.
   - Example: CSSE 497-498-499 (Senior Projects I-III) Test Results Report
   - Not acceptable: Insufficient testing planned or executed.
5. Document the finished product or process as appropriate for the discipline according to standard practice.
   - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) User Documentation
   - Description: The student should provide instructions in the use and maintenance of the product or process.
   - Example: CSSE 497-498-499 (Senior Projects I-III) User Documentation
   - Not acceptable: Insufficient documentation provided
6. Present and transfer the product or process and documentation to the client.
   - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Project presentation at expo
   - Description: The student should prepare a description of the finished product or process for display at the Senior Project Expo
   - Example: CSSE 497-498-499 (Senior Projects I-III) Expo Poster
   - Not acceptable: Poorly-designed or incoherent description of project results.

**Senior Exit Interviews**

Each computer science student who is eligible to participate in the annual commencement exercises at the end of the academic year is invited to participate in a one-on-one exit interview with the Head of Computer Science and Software Engineering. At the end of AY 2005-2006, 19 students participated in the exit interviews.

The questions discussed during the interviews were as follows:

1. What is the best thing that has happened to you as a computer science major?
2. What is the worst thing that has happened to you as a computer science major?

3. Please make any comments you think appropriate regarding any faculty member in the department.
4. In your opinion, what are the major strengths in the computer science curriculum?
5. In your opinion, what is the most significant weakness in the department's curriculum?
6. What were the most important classes you took?
7. What classes did you enjoy the most?
8. What were your least favorite classes?
9. What is the most important change needed in the department?
10. Rate how well you feel you have satisfied each of the following CS program outcomes on a scale of 1 (low) to 5 (high):

    a. Apply effectively a variety of programming languages, programming paradigms, operating systems, networks, and software development tools.

    b. Anticipate complexities and problems involved in the development of large software systems.

    c. Analyze requirements, design software that satisfies those requirements, and implement that software.

    d. Analyze problems using ideas of problem complexity, models of computation, and decidability.

    e. Design algorithms using a variety of paradigms

    f. Analyze algorithms in terms of correctness, as well as time and space efficiency.

    g. Communicate effectively, both verbally and in writing.

    h. Evaluate and discuss the legal, social, and ethical aspects of significant events that arise in the computing industry.

    i. Collaborate effectively in small teams.
11. Did you do a senior thesis or project?
12. What are your post graduation plans?
13. 13. Any closing comments?

**Alumni Surveys**

In August 2005, the dean and department heads of Rose-Hulman Institute of Technology agreed to develop the first Institute-wide Alumni Survey. The purpose of the survey was to collect information from alumni for use in various reports, including the ABET Self Study Reports (for program accreditation) and the North Central Association /AQIP Systems Portfolio (for institutional accreditation).

The survey instrument was developed with assistance from the staff of the Office of Institutional Research, Planning, and Assessment. The IRPA staff first examined standard alumni survey instruments to determine potential question formats and types. A draft of the Institutional Survey was then submitted to department heads and the dean for input. The Institutional Survey included questions that were common among all departments, including questions about demographics, current employment, continuing education, and Institutional learning outcomes. Some departments, however, submitted program-specific questions; these questions were related to

Program Educational Objectives and Program Educational Outcomes relevant to ABET accreditation. The following departments provided department-specific items for the survey: Applied Biology/ Biomedical Engineering; Applied Optics/ Optical Engineering; Chemical Engineering; Civil Engineering; Computer Science/Software Engineering; and Mathematics. All other departments used the Institutional version of the survey without additional items added.

The survey was published online by IRPA staff on October 14, 2005. In order to invite participation, department heads sent email and/or a letter to their program's alumni asking them to complete the survey. Email addresses were obtained through the RHIT Alumni Office. Some departments included a reference to the survey, as well as the URL, on their departmental homepages or newsletter. Given the large number of alumni who have graduated from some programs, IRPA encouraged the use of a statistical sample of alums, rather than asking all alumni to participate. In addition to the statistical sampling, alumni who were 5, 10, 15, and 20 years from graduation were targeted. This again was in an effort to narrow the pool of respondents while still ensuring quality data collection . The survey remained open until December 15, 2005.

**CSSE Board of Advisors**

The CSSE Board of Advisors is composed of leaders from the private and academic sectors with considerable expertise in software and computing. It intentionally includes several alumni representatives. The board nominally meets once each spring and once each fall, with at least one of those meetings being on campus. Board meetings are attended by as many faculty as possible, and follow agendas similar to the following:

- call to order and introductions
- report on the status of the department
- visit to a class
- discussion with students without faculty present
- discussion of what it sees as emerging technologies of which the department should be aware
- special topics (e.g. accreditation)
- meetings with the Dean of Academic Affairs and/or the President of the Institute
- executive session
- wrap-up session

The report on the status of the department includes noteworthy items from the current computer science program assessment report. In particular, it includes discussion of the extent to which the program is achieving the POs and PEOs, as well as any proposed changes to the POs or PEOs. The business transacted during the board meetings is captured in the minutes of the meeting, and the board provides the department with a follow-up report including its assessment of the state of the department as well as any recommendations that it has.

***Aspects of CIP Most Relevant to CAC Accreditation Criteria***

Table 7 summarizes the following information regarding the various assessment mechanisms and how they are used in the CIP for the computer science program: frequency and timing of assessments, what data are collected, how data are collected, from whom data are collected, and how assessment results are used and by whom. The actual documentation that was generated during AY 2005-2006 is included as Appendix VII. Program Assessment Data.

**Table 7. Aspects of Continuous Improvement Process most relevant to CAC accreditation criteria.**

| | Frequency and Timing | Data collected | How data are collected | From whom | How assessment results are used and by whom |
|---|---|---|---|---|---|
| Course Assessment | After each course offering | Actions taken to address previous recommendations<br>Other changes to course<br>Extent of achievement of COs<br>Instructors' qualitative assessments<br>Relevant student feedback<br>Recommendations for next offering | Data captured from student performance on graded work<br>Summary results reported in Course Assessment Report | Course Coordination Teams | Future instructors to evaluate extent to which course is achieving COs |
| Portfolio Assessment | Each summer | Extent of achievement of ILOs | Faculty pairs rate on the basis of a rubric and make "Yes/No" determinations | IRPA | Department faculty to evaluate extent to which program is achieving ILOs |
| Senior Exit Interview | Near the end of each academic year | Satisfaction with POs | Department head interviews each willing graduating senior one-on-one | Seniors | Department faculty to evaluate extent to which program is achieving POs |
| Alumni Survey | Each academic year | Demographic information<br>Education<br>Employment<br>Preparation for success<br>Value of degree | Web-based survey | Alumni | Institute faculty to evaluate appropriateness of ILOs<br>Department faculty to evaluate extent to which program is achieving ILOs |

| | | | | | |
|---|---|---|---|---|---|
| | | Likelihood of recommending program<br>Overall satisfaction with education<br>Curricular recommendations<br>Adequacy of contact with alumni<br>Program strengths<br>Program weaknesses<br>Program opportunities<br>General comments<br>Perceived importance of ILOs<br>Perceived contribution of programs to achievement of ILOs | | | |
| Board of Advisors | At least annually | Emerging trends, practices, and technologies<br>Feedback on adequacy of PEOs and POs<br>Feedback on proposed changes to program | Business of meeting recorded in minutes<br>Board produces follow-up report | Board of Advisors | Department faculty to evaluate appropriateness of POs and PEOs and proposed program changes |

## Program Improvements

There has been a steady evolution of the computer science curriculum over the course of its existence. Languages, operating systems, and hardware used in some courses have changed as frequently as every three years. Upper level elective offerings have changed in response to faculty and student interests. Faculty members have consulted the ACM curriculum recommendations as well as the ABET CAC and EAC accreditation requirements when determining major requirements and department course offerings.

Computer science degrees have been awarded at Rose-Hulman since 1971, when the department was part of the Division of Electrical Engineering and Computer Science. From 1970 to 1987 the computer science major placed nearly equal emphasis on computer science and electrical engineering. In 1987 the Department of Computer Science became a completely independent academic unit, and the next year a new major curriculum became effective. This new curriculum significantly increased the number of required computer science and mathematics courses. An increased emphasis on software engineering concepts was also added to the curriculum.

### *Major Program Changes Within the Last Five Years*

The most significant development made in our program within the last five years has been a significant increase in the visibility of the software engineering component. These changes have been driven primarily by two facts. First, today's science, engineering, and mathematics students will almost certainly be involved in at least one significant software development project during their career. As such, every Rose-Hulman student should receive at least some education in software engineering as fundamental subject matter within computer science. Second, the vast majority of our computer science graduates begin their careers in software engineering positions. As such, software engineering is an important discipline in its own right.

In recognition of the growing importance of software engineering, the department became the Department of Computer Science and Software Engineering in 2001. At the recommendation of the department faculty, an interdisciplinary committee was formed to develop a curriculum for an undergraduate major in software engineering. The Institute faculty and the Board of Trustees approved the new degree program in the spring of 2003. The first student to earn the software engineering major graduated in 2004.

The emphasis on software engineering is still very much present in the computer science program, both in the Fundamentals of Software Development sequence and in the two required Software Engineering courses. However, the creation of a separate Software Engineering program has given us the freedom to increase the theoretical content of the computer science program, which we are in the process of doing.

The department also moved into newly renovated space in 2003, greatly expanding our office and laboratory areas.

### *Significant Future Program Improvement Plans*

Numerous improvements have been made to courses as a result of the course assessment process, which will be documented in the Program Assessment Report and made available to the CAC Team Chair and Program Evaluator when it is completed later this summer.

### *Changes in PEOs and POs*

There have been no significant changes to the PEOs and POs, since they were approved for the first time during AY 2005-2006. The department faculty will officially consider adding a PO related to ongoing professional development at the beginning of AY 2006-2007.

## D. Documentation

The purpose of this section is to address the following question:

> Is there appropriate documentation of the program educational objectives, the program outcomes, and the assessment and continuous improvement processes?

Table 8 summarizes the documentation of the PEOs, POs, CIP, and assessment reports. The URL of the department website is http://www.cs.rose-hulman.edu/. The site is viewable by anybody with access to the World Wide Web, and in particular to all program constituents. The URL of the department TWiki site is http://twiki.cs.rose-hulman.edu. The TWiki site is intended to be viewable only by department faculty, staff, and students. The parts of the site relevant to assessment results are intended to be editable only by department faculty and staff.

**Table 8. Documentation of PEOs, POs, CIP, and assessment reports.**

| | Documentation | Who has access |
|---|---|---|
| Measurable PEOs | Department website<br>Future undergraduate bulletin | Anyone with WWW access |
| Program outcomes | Department website<br>Future undergraduate bulletin | Anyone with WWW access |
| Assessment process | Department TWiki site | Department faculty, staff, and students |
| Results of assessment | Department TWiki site | Department faculty, staff, and students |

## 2. Student Support

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> Students can complete the program in a reasonable amount of time. They have ample opportunity to interact with their instructors. Students are offered timely advising, by qualified individuals, about the program's requirements and their career alternatives. Students who graduate from the program meet all program requirements.

### A. Frequency of Course Offerings

All required courses are offered at least once per year, with some courses offered twice a year and two of the fundamentals courses offered every term (three times a year). Also, all of the department's electives, listed in Table 9, are offered annually.

**Table 9. CSSE courses that can be used to fulfill elective requirements of the Computer Science program.**

| Dept Course # | Title of course | Quarter Hours |
|---|---|---|
| CSSE 325 | Fractals and Chaotic Dynamical Systems | 4 |
| CSSE 351 | Computer Graphics | 4 |
| CSSE 373 | Formal Methods in Specification and Design | 4 |
| CSSE 374 | Software Architecture and Design I | 4 |
| CSSE 375 | Software Construction and Evolution | 4 |
| CSSE 376 | Software Quality Assurance | 4 |
| CSSE 377 | Software Architecture and Design II | 4 |
| CSSE 404 | Compiler Construction | 4 |
| CSSE 413 | Artificial Intelligence | 4 |
| CSSE 432 | Computer Networks | 4 |
| CSSE 442 | Computer Security | 4 |
| CSSE 451 | Advanced Computer Graphics | 4 |
| CSSE 453 | Topics in Artificial Intelligence | 4 |
| CSSE 461 | Computer Vision | 4 |
| CSSE 473 | Design and Analysis of Algorithms | 4 |
| CSSE 474 | Theory of Computation | 4 |
| CSSE 479 | Cryptography | 4 |
| CSSE 481 | Web-Based Information Systems | 4 |
| CSSE 490 | Special Topics in Computer Science | 4 |

### B. Interaction with Faculty

Rose-Hulman faculty, and CSSE faculty in particular, pride themselves on close interaction with our students. Our faculty are considered coaches and mentors rather than the more traditional teachers. All of the department faculty are on a first-name basis with our students. Many of our courses have a formal laboratory component, and our faculty are required to be in the lab with their students, thus facilitating close faculty-student interaction. Department faculty maintain liberal office hours, allowing students plenty of opportunities for outside of class help. Most faculty give students their home phone number and/or instant messenger username so students can get hold of their teachers after hours as well. Most department faculty also serve as active advisors to several student clubs and organizations on campus, allowing students and faculty to get to know each other outside of the classroom and in a setting where they share a common hobby or interest.

## C. Student Advising

### Freshman Adviser Program

The Office of Student Affairs administers a formal advising program for all first-year students. Students who have declared an interest in majoring in biomedical, civil, software, or optical engineering are matched with faculty advisors from their departments. This matching occurs because the number of majors in those departments is small and students can be advised by 2-4 faculty. All students majoring in computer science are assigned a faculty advisor in the department, with the exception of those who attended the Fast Track Calculus program in the summer prior to their first year on campus; they are assigned a Fast Track Calculus instructor as an advisor for their freshman year. Students who plan to major in large departments like electrical and computer engineering, or students who have not expressed interest in a particular program are given a faculty advisor from among the volunteers from all the departments on campus: humanities and social sciences, math, chemistry, physics, etc. Approximately 30 faculty volunteer to work as Freshman Advisors each year; the faculty/student ratio is about 16:1.

All faculty advisors are required to participate in a half-day training session prior to the start of each academic year. During the Freshman Advisor Workshop, faculty learn about the following topics:

- Their responsibilities as an advisor (tasks they must complete, forms they must help students fill out, etc.)
- Information about Rose-Hulman student retention and attrition
- Purpose and frequency of advisor-advisee meetings and activities (advisors meet with each advisees 4 times during the quarter)
- Information about the Rose E-Portfolio (used for collecting evidence of student learning outcomes)
- Purpose and content of the first-year course “College and Life Skills,”
- Availability of academic and counseling services
- Procedures for registering students for courses.

Roughly ½ of the Workshop time is spent on instructing advisors in the course registration procedures. All these topics are also included in an advisor manual written and distributed by the Office of Student Affairs.

The process of registration for first-year students in the fall quarter begins before they arrive on campus. During the summer, first-year students are pre-registered for courses by the Registrar’s office. Students are assigned courses based on their intended majors according to the curricula of

the appropriate department. After students arrive on campus, they meet with their faculty Freshman Advisors; at that meeting, advisors double-check the pre-registered schedules, make changes for students who want to switch from one intended major to another, help undecided students choose a tentative major, and assist students who arrive with advanced placement credits to select alternate courses, such as electives in the humanities and social sciences. Advisors then assist students with the on-line registration process to ensure that each student has a full-time, non-overloaded schedule that meets their needs. Problems with registration or requests for overload schedules are handled by the Registrar's office.

The support provided by Freshman Advisors doesn't end when course registration is over. The Office of Student Affairs provides funds for advisors to host social gatherings, like a pizza party, for their advisees. The purpose of such informal gatherings is to encourage students to look upon their Freshman Advisors as faculty they can talk to and be comfortable with. Such relationships can provide support to students when they encounter problems during their first year at Rose-Hulman.

Support is also evident in the department presentations that are given approximately one month after classes begin. All first-year students are required to attend an evening of program presentations during which faculty representing each degree program at Rose-Hulman give one-hour presentations on the curriculum, degree requirements, and subsequent career opportunities. The purpose of the presentation is to provide students with realistic insights into what happens in the major and what the courses are like. Each program is presented twice so that students may attend presentations for both a first- and a second-choice degree area. From these presentations, students may decide to switch majors or even add a minor or second degree that reflects his/her interests.

Throughout the first year, regular contact between students and Freshman Advisors is ensured by Rose-Hulman's course registration procedures, which is described below. After the first year, student advising procedures are handled through individual departments or programs, with assistance from the Registrar's Office and Student Affairs as needed.

**Course Registration Procedures**

The Registrar's office assigns an identification number to each student prior to registration for each quarter (each student's number changes each quarter). Without these numbers students cannot access the on-line registration system and are therefore unable to register for classes. The identification numbers are distributed to faculty advisors, who then meet with advisees to give out the identification numbers for that quarter, to check registration plans, and to discuss the students' academic progress. To additionally facilitate course planning and academic advising, students' midterm grades are released to academic advisors before registration for the next quarter begins. This helps academic advisors monitor student progress. Finally, the Rose-Hulman catalog is online and is updated annually. The catalog lists all requirements for the major, a suggested course layout to complete the major in four years, and descriptions of all courses offered.

**Career Choices**

All computer science faculty are eager and willing to talk with any computer science student at Rose-Hulman about course decisions and career choices, whether or not that faculty member serves as the student's academic advisor. All faculty maintain an open door when in their office – students are welcome to drop in at any time. Students may also contact a faculty member by e-mail or phone to setup an appointment.

## D. Meeting the Requirements

The Office of the Registrar, under the direction of Mr. Tim Prickel, is the primary repository of student academic records. The Registrar and the Associate Registrar, Ms. Jan Lind, ensure that student records are both current and correct. A file is maintained for each student. The file contains the mid-term and final grade reports for each academic quarter; these reports show how the student is performing in each course. The file also contains other pertinent academic information, including memoranda from the Registrar regarding academic warnings, academic probation, and/or failure to maintain the GPA-in-major requirement. There are also program checklists so the Registrar can track when a student fulfills each requirement (both for the department and for the Institute). The student file maintained by the Registrar's Office is the definitive record that charts a student's progress through the four-year program. The Office of the Registrar follows the Institutional Academic Rules and Procedures Manual; this manual is available to students both on line and in printed form from the Office of Student Affairs.

In order to monitor student completion of academic requirements, the following procedure is used. During the freshman, sophomore, and junior years, the primary responsibility for tracking student progress lies with the student's Freshman Advisor (first year) or Academic Advisor (subsequent years). The advisor keeps a Program Checklist in order to track the student's completion of required courses for the major, as well as all Institute graduation requirements. Prior to the fall of the student's senior year, the Registrar's Office reviews the file of each rising senior to confirm that the student is on track for graduation. If the student has failed to take a required course or to fulfill the appropriate number and/or distribution of Humanities and Social Sciences courses, the student is notified by the Registrar's Office. This gives the student three quarters by which to complete the necessary courses and requirements. Through the senior year, the student is tracked by the Registrar's Office until the Final Degree Audit, a process by which all students' requirements are checked and verified. Finally the entire faculty approves all students for graduation during a meeting of the faculty. At that time, all students are approved by a vote of the faculty.

The Office of the Registrar provides other services for faculty and students at Rose-Hulman. The office has an excellent track record in arranging the schedule of course offerings so that courses required for a given program will not be offered at conflicting times. The Office also ensures that students who must take a required course at a particular time are accommodated before students whose schedules are a bit more flexible. In addition, the staff of the office provides support to the academic advisors. During the Freshman Advisor Orientation, the Registrar gives a presentation familiarizing advisors with new or existing policies. Guidance is also offered on an individual basis to faculty who require more information on how to access students schedule information, rules and procedures, and on-line course registration.

## 3. Faculty Qualifications

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> Faculty members teaching in the program are current and active in the associated computing discipline. They each have the educational backgrounds or expertise consistent with their expected contributions to the program. Each has a level of competence that normally would be obtained through graduate work in the discipline, relevant experience, or relevant scholarship. Collectively, they have the technical breadth and depth necessary to support the program.

### A. Faculty Profile

All faculty are full time and hold Ph.D.'s in the fields indicated in Table 10.

**Table 10. Faculty profile (name, academic rank, Ph.D. field, research areas, and advisee count)**

| Faculty Name | Academic Rank | Ph.D. Discipline | Research Areas | # Advisees |
|---|---|---|---|---|
| Cary Laxer | Professor and Head | Biomedical Engineering | Computer graphics | 18 |
| Claude Anderson | Professor | Mathematics | Programming languages, compilers, systems programming | 0[11] |
| Mark Ardis | Professor | Computer Science | Software engineering, product line engineering, formal methods | 19 |
| Don Bagert | Professor | Computer Science | Software engineering, software process improvement, software methodologies | 18 |
| Matt Boutell | Assistant Professor | Computer Science | Multimedia content understanding, computer vision, pattern recognition | 0[12] |
| Steve Chenoweth | Associate Professor | Computer Science and Engineering | Software engineering, software architecture | 21 |
| Archana Chidanandan | Assistant Professor | Computer Engineering | Hardware design, image processing, wireless communications | 22 |
| Curt Clifton | Assistant Professor | Computer Science | Programming languages, compilers, aspect-oriented programming | 0[13] |
| Lisa Kaczmarczyk | Assistant Professor | Interdisciplinary | Cognitive science, artificial intelligences, computer science education | 0[14] |
| J.P. Mellor | Associate Professor | Electrical Engineering, Computer Science | Computer vision, human-computer interaction | 0[15] |
| Larry Merkle | Assistant Professor | Computer Engineering | Evolutionary computation, computational science and engineering, computer security | 21 |
| David Mutchler | Professor | Computer Science | Artificial intelligence, machine game-playing, probabilistic analysis of algorithms | 22 |
| Michael Wollowski | Associate Professor | Computer Science | Diagrammatic logic, World Wide Web, artificial intelligence | 14 |

## B. Information Regarding Faculty Members

The following pages provide faculty curriculum vitae for each member of the department faculty.

[11] on sabbatical in 2005-6

[12] new faculty member in 2005-6

[13] new faculty member in 2005-6

[14] new faculty member in 2005-6

[15] on sabbatical n 2005-6

***Faculty Curriculum Vitae: Cary Laxer***

**Academic Rank and Title:**

Professor and Head of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | September, 1981, Assistant Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | September, 1984, Associate Professor<br>September, 1993, Professor |
| Years of Service at Rose-Hulman: | 25 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1980 | Biomedical Engineering | Duke University |
| B.A. | 1976 | Computer Science and Mathematics | New York University |

**Professional Development Activities (AY 2001-2006):**

- ACM Member: SIGCSE, SIGGRAPH
- IEEE Senior Member: Computer Society
- ASEE Member
- Registrar, ACM SIGCSE ITiCSE Conference, 2004, 2006.
- Registrar, ACM SIGCSE Technical Symposium, 1996-date.
- Attended SIGCSE, ITiCSE, and SIGGRAPH conferences annually.
- Reviewer for SIGCSE, ITiCSE, and CGEMS.
- Co-chaired ACM SIGGRAPH Education Committee workshop on defining a computer graphics curriculum, 2003-date
- Program chair for CGE04, Computer Graphics Educators Workshop, Zhejiang University, Hangzhou, China
- Participant in CGE02, Computer Graphics Educators Workshop, University of Bristol, Bristol, England

**Professional Experience:**

| | |
|---|---|
| 1981 – Present | Assistant Professor/Associate Professor/Professor<br>(Department Head since 2002)<br>Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1999 – 2000 | Visiting Professor<br>Department of Information Systems and Computing<br>UNITEC Institute of Technology, Auckland, New Zealand |
| 1989 – 1990 | Lilly Visiting Professor of Emerging Cardiovascular Technologies<br>Duke University-University of North Carolina NSF Engineering<br>Research Center, Duke University, Durham, NC |

**Consulting Activities:**

- University of Virginia's College at Wise – consulted on development of new computer science and software engineering programs, 2003.
- Harvey Mudd College – external evaluator of computer science program, 2006.

**Assigned Duties (AY 2005-2006):**

- Curriculum Committee - 2 hours/week
- Academic and student organization advising - 3 hours/week
- Administrative Council - 1 hour/week
- Greek Advisory Committee - 2 hours/term
- Wellness Committee - 2 hours/term

**Institutional Service (AY 2001-2005):**

- Member of Faculty Affairs Committee 2000-2002; chair 2001-2002.
- Member of Curriculum Committee, 2002-date.
- Member of Fraternity Advisory Council, 1988-date.
- Chapter Advisor to Triangle Fraternity, 1988-date.
- Faculty advisor to approximately 20-25 students majoring in computer science, annually.

**Principal Publications (AY 2001-2006):**

- C. Laxer and A. Young. "Quality Assurance: How much is needed?" Proceedings of the 13th Annual Conference of the NACCQ, 179-182, 2000.
- C. Laxer. "Using Computer Graphics to Reinforce Mathematics." Proceedings of the 17th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Short Papers and Works in Progress, 205-209, 2000.
- C. Laxer. "Treating Computer Science as Science: An Experiment with Sorting." Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, 189, 2001.
- M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y.B. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz. "Report by the ITiCSE 2001 Working Group on Assessment of Programming Skills of First-year CS Students." SIGCSE Bulletin, 33:4, 125-140, 2001.
- C. Laxer. "Evaluating Students Team Project Experiences." Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, 198, 2002.
- C. Laxer. "A Student Team Research Approach to the Second Course in Computer Graphics." Proceedings of CGE02 Eurographics - ACM SIGGRAPH Workshop on Computer Graphics Educations, 1-3, 2002.
- M. Dick, J. Sheard, C. Bareiss, J. Carter, D. Joyce, T. Harding, and C. Laxer. "Addressing Student Cheating: Definitions and Solutions," SIGCSE Bulletin, 35:2, 172-184, 2003.
- S. Alexander, J. Amillo, R. Boyle, M. Clark, M. Daniels, C. Laxer, K. Loose, and D. Shinners-Kenndy. "Case Studies in Admissions to and Early Performance in Computer Science Degrees," SIGCSE Bulletin, 35:4, 137-147, 2003.

- S. Cunningham, W. Hansmann, C. Laxer, and J. Shi. "The Beginning Computer Graphics Course in Computer Science," Computer Graphics, 38:4, 24-25, 2004.
- M. Daniels, L. Barker, A. Cajander, C. Laxer, and D. Moore. "Managing Cross-Cultural Differences in an Open Ended Group Project Course," Proceedings of 35th ASEE/IEEE Frontiers in Education Conference, 2005.
- U. Fuller, J. Amillo, C. Laxer, W.M. McCracken, and J. Mertz. "Facilitating student learning through study abroad and international projects," SIGCSE Bulletin, 37:4, 139-151, 2005.

**Selected Honors and Awards:**

- Recipient of Dean's Outstanding Teacher Award, Rose-Hulman Institute of Technology, 1987
- President's Outstanding Faculty Member of the Week, Rose-Hulman Institute of Technology, 1990, 1992 (twice), 1993 (twice), 1995, 1996 (three times), 1998, 1999, 2000, 2001, 2002 (twice), 2003, 2004 (twice)
- ACM Recognition of Service Award, 1990, 1993 (twice), 1995, 1996, 2000, 2001
- Member of Pi Mu Epsilon, Upsilon Pi Epsilon, and Blue Key honorary societies
- Life Member of Alpha Phi Omega national service fraternity (Tau Lambda Chapter Distinguished Service Key, 1991; National Distinguished Alumnus Key, 1998)
- Listed in Outstanding Young Men of America, Who's Who in Frontier Science and Technology, Who's Who Among America's Teachers, American Men and Women of Science
- Member of European Academy of Sciences
- Named Teacher of the Year by Triangle Fraternity, 1984, 2004
- Honorary Alumnus Award, Rose-Hulman Alumni Association, 2002

**Other Scholarly Activity (AY 2001-2006):**

- Co-chair of computer graphics curriculum development efforts of the ACM SIGGRAPH Education Committee

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE351-01 | Computer Graphics | 4 | 30 |
| Fall 2004-2005 | CSSE491-02 | Directed Independent Studies | 4 | 4 |
| Winter 2004-2005 | CSSE451-01 | Advanced Computer Graphics | 4 | 9 |
| Spring 2004-2005 | CSSE490-02 | The History of Computing | 4 | 19 |
| Spring 2004-2005 | CSSE494-01 | Senior Thesis I | 4 | 1 |
| Fall 2005-2006 | CSSE351-01 | Computer Graphics | 4 | 34 |
| Fall 2005-2006 | CSSE490-01 | Topics in Computer Science | 4 | 4 |
| Fall 2005-2006 | CSSE494-02 | Senior Thesis I | 4 | 1 |
| Fall 2005-2006 | CSSE495-01 | Senior Thesis II | 4 | 1 |
| Winter 2005-2006 | CSSE451-01 | Advanced Computer Graphics | 4 | 21 |

| | | | | |
|---|---|---|---|---|
| Winter 2005-2006 | CSSE495-02 | Senior Thesis II | 4 | 1 |
| Winter 2005-2006 | CSSE496-01 | Senior Thesis III | 4 | 1 |
| Spring 2005-2006 | CSSE325-01 | Fractals & Chaotic Dynam Syst | 4 | 12 |
| Spring 2005-2006 | CSSE351-01 | Computer Graphics | 4 | 1 |
| Spring 2005-2006 | CSSE496-02 | Senior Thesis III | 4 | 1 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 15%**

- I have been leading the efforts to establish a computer graphics curriculum for the ACM SIGGRAPH Education Committee.
- I have co-chaired workshops on this effort at the past three SIGGRAPH conferences.
- I have also been active in promoting global opportunities for computer science students, and served on a working group to address that issue at the 2005 ITiCSE conference in Lisbon, Portugal.

***Faculty Curriculum Vitae: Claude Anderson***

**Academic Rank and Title:**

Professor of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | September, 1988, Associate Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | September, 1994, Full Professor |
| Years of Service at Rose-Hulman: | 18 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Masters | 1987 | Computer Science | Indiana University (Bloomington) |
| Ph.D. | 1981 | Mathematics | University of Illinois (Urbana) |
| Masters | 1977 | Mathematics | University of Illinois (Urbana) |
| B.S. | 1975 | Mathematics | Caltech |

**Professional Development Activities (AY 2001-2006):**

- Member, Association for Computing Machinery (ACM)
- Spent my 2005-06 sabbatical learning (by doing) about large-scale web development, .NET, databases, UI design, quality assurance.
- Attended SIGCSE 2004
- Attended OOPSALA 2002 and participated in the Educators' Symposium.
- Attended Microsoft TechEd 2001.
- Attended the ANGEL Users conferences in 2004, 2005, and 2006.
- Reviewed approximately 15 textbooks for various publishers.

**Professional Experience:**

- 1988 – Present Assoc. Professor/Professor
  Computer Science and Software Engineering Department
  Rose-Hulman Institute of Technology, Terre Haute, IN
- 2005 – 2006 Software Engineer
  ANGEL Learning, Inc., Indianapolis, IN
- Summer 1998 Software Engineer
  Beckman Instruments, Indianapolis, IN
- 1996 – 1997 Software Engineer
  Beckman Instruments, Indianapolis, IN
- Summers 1994 – 95 Software Engineer
  Navistar Technical Center, Fort Wayne, IN
- 1981 – 1988 Assistant Professor, Mathematics and Computer Science
  Wilkes College, Wilkes-Barre, PA

**Consulting Activities:**

- Software design and development, Pennsylvania Funeral Directors Association.

**Assigned Duties (AY 2005-2006): None**

**Institutional Service (AY 2001-2005):**

- Chair, RHIT Banner Web faculty Team, 2001-03
- Chair, RHIT Course Management System evaluation and Selection Team, 2003-04
- Leader, RHIT Implementation, Education, and Support Team, 2004-05
- Freshman and New Grad Student Computer Orientation coordinator, 1999-2004
- RHIT Leaves Committee, 2002-2004
- Employee Relations Committee, 2001-02
- Quality of Education Committee, 2000-2001
- Electronic Portfolio Rater, 2001-2003
- Operation Catapult Faculty, 2001-2004
- Selection Committee for Computing Center Director, 2001-2002
- Served on a Math Department team to design and standardize the Discrete Math sequence, 2002-03.
- Led an inter-departmental team that examined the math requirements for CS majors, 2003-04.
- Part of the design teamed that planned our new intro courses, did the first implementations of 220 and 230.

**Principal Publications (AY 2001-2006):**

- "Transitioning to an Objects-Early Three-Course Introductory Sequence: Issues and Experiences" proceedings of the SIGCSE Computer Science Education Conference, Norfolk, VA, February, 2004.

**Selected Honors and Awards:**

- Carpenter Outstanding Teacher Award, Wilkes College, 1986

**Other Scholarly Activity (AY 2001-2006):**

- Sabbatical at ANGEL Learning, which has been fully devoted to software design and development, with a touch of customer relations thrown in

**Courses taught (AY 2003-2005):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2003-2004 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 28 |
| Fall 2003-2004 | CSSE304-02 | Programming Language Concepts | 4 | 26 |
| Fall 2003-2004 | CSSE304-03 | Programming Language Concepts | 4 | 21 |
| Winter 2003-2004 | CSSE230-01 | Fundamntls of Software Dev III | 4 | 31 |
| Winter 2003-2004 | CSSE230-02 | Fundamntls of Software Dev III | 4 | 28 |
| Spring 2003-2004 | CSSE304-01 | Programming Language Concepts | 4 | 18 |

| | | | | |
|---|---|---|---|---|
| Spring 2003-2004 | CSSE304-02 | Programming Language Concepts | 4 | 22 |
| Spring 2003-2004 | CSSE304-03 | Programming Language Concepts | 4 | 20 |
| Fall 2004-2005 | CSSE404-01 | Compiler Construction | 4 | 21 |
| Fall 2004-2005 | CSSE473-01 | Design & Anlsys of Algorithms | 4 | 14 |
| Fall 2004-2005 | CSSE494-01 | Senior Thesis I | 4 | 1 |
| Winter 2004-2005 | CSSE230-01 | Fundamntls of Software Dev III | 4 | 18 |
| Winter 2004-2005 | CSSE230-02 | Fundamntls of Software Dev III | 4 | 26 |
| Winter 2004-2005 | CSSE495-01 | Senior Thesis II | 4 | 1 |
| Spring 2004-2005 | CSSE230-01 | Fundamntls of Software Dev III | 4 | 17 |
| Spring 2004-2005 | CSSE230-02 | Fundamntls of Software Dev III | 4 | 13 |
| Spring 2004-2005 | CSSE496-01 | Senior Thesis III | 4 | 1 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 5%**

- Mainly I have been learning about new technologies and trying to keep courses up-to-date based on what I have learned.

***Faculty Curriculum Vitae: Mark A. Ardis***

**Academic Rank and Title:**

Professor of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | September, 2000, Professor (untenured) |
| Date(s) of Promotion(s) and Resulting Rank(s): | September, 2005, Tenured Professor |
| Years of Service at Rose-Hulman: | 6 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1980 | Computer Science | University of Maryland |
| Masters | 1977 | Computer Science | University of Maryland |
| B.S. | 1971 | Mathematics | Cornell University |

**Professional Development Activities (AY 2001-2006):**

- Member, Association for Computing Machinery (ACM)
- Member, IEEE Computer Society
- Member, American Society for Engineering Education (ASEE)
- Managed software engineering projects at Rose-Hulman Ventures
- Co-chair of interdisciplinary conference on the World Wide Web at Rose-Hulman (WWW@10)
- Contributed software engineering education materials to online library at SWENET Workshop, 2005
- Published articles (and presentations) in software engineering education

**Professional Experience:**

| | |
|---|---|
| 2000 – Present | Professor<br>Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1991 – 2000 | Member of Technical Staff<br>Bell Labs, Naperville, IL |
| 1987 – 1991 | Senior Computer Scientist<br>Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA |
| 1983 – 1987 | Associate Professor of Information Technology<br>Wang Institute of Graduate Studies, Tyngsboro, MA |
| 1980 – 1983 | Assistant Professor of Computer Science<br>University of Illinois, Urbana, IL |
| 1974 – 1976 | Member of Technical Staff<br>Delta Research Corporation, Arlington, VA |
| 1972 – 1974 | Consultant<br>Macro Systems Incorporated, Silver Spring, MD |

1971 – 1972 Associate Member of Technical Staff
Computer Sciences Corporation, Silver Spring, MD

**Consulting Activities:**

- 2000 - 2002 Innovation Fellow, Rose-Hulman Ventures, Terre Haute, IN

**Assigned Duties (AY 2005-2006):**

- Commission on the Assessment of Student Outcomes (1 hr/wk)

**Institutional Service (AY 2001-2005):**

- General Co-Chair and Program Chair for WWW@10 Conference, 2004
- CCSE Foundations Knowledge area for Computing Curricula 2001
- Team Leader for IEEE Certified Software Development Professional Exam
- Reviewer for ACM, ASEE and IEEE conferences and journals
- Chair of Search Committee for CSSE Department Head, Rose-Hulman Institute
- Secretary of Academic Computing Committee, Rose-Hulman Institute
- Chair of Advisory Committee on Computer Use, Rose-Hulman Institute
- Member of Faculty Affairs Committee, Rose-Hulman Institute
- Reviewer of ABET Accreditation Materials, Rose-Hulman Institute

**Principal Publications (AY 2001-2006):**

- Proceedings of IFIP WG8.6 Working Conference on Diffusing Software Product and Process Innovations, M. Ardis and B. Marcolin, (editors), Kluwer Academic Publishers, 2001.
- "Software Engineering Baccalaureate Programs in the United States: an Overview", D. Bagert and M. Ardis, 33rd ASEE/IEEE Frontiers in Education Conference, November 2003.
- "Automating the Process of Assigning Students to Cooperative-Learning Teams", R. Cavanaugh, M. Ellis, R. Layton, and M. Ardis, 2004 American Society for Engineering Education Annual Conference & Exposition, June 20-23, 2004.
- "Test-First Teaching: Extreme Programming Meets Instructional Design in Software Engineering Courses", M. Ardis and C. Dugas, 34th ASEE/IEEE Frontiers in Education Conference, October 20-23, 2004.
- "An Incomplete History of Master of Software Engineering Programs in the United States", 15th Reunion of CMU MSE Program, August 6, 2005.
- "Diversity of Interaction in a Quality Assurance Course", M. Ardis and C. Dugas, 35th ASEE/IEEE Frontiers in Education Conference, October 19-22, 2005.

**Selected Honors and Awards:**

- First recipient of Nancy Martin Award for Excellence in Teaching, Wang Institute, 1985.
- First recipient of Excellence in Teaching Award, CMU Master of Software Engineering Program, 1991.
- PAVE Award for Technology Transfer Symposium, Lucent Technologies, 1996.
- PAVE Award for Technology Transfer Symposium, Lucent Technologies, 1998.

- PAVE Award for Contributions to Domain Engineering, Lucent Technologies, 1998.

**Other Scholarly Activity (AY 2001-2006):**

- Patent 6,580,439: (with Robert Olsen and Paul Pontrelli) Graphical user interface for providing component relational information for a uniform configuration controller for replicated component systems, 2003.
- Patent 6,591,373: (with Robert Olsen and Paul Pontrelli) Uniform configuration controller for replicated component systems, 2003.

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE371-01 | Software Requirements and Spec | 4 | 13 |
| Fall 2004-2005 | CSSE371-02 | Software Requirements and Spec | 4 | 25 |
| Fall 2004-2005 | CSSE372-03 | Software Requirements and Spec | 4 | 24 |
| Fall 2004-2005 | CSSE373-01 | Formal Methods in Spec & Desn | 4 | 11 |
| Winter 2004-2005 | CSSE372-01 | Software Project Management | 4 | 23 |
| Winter 2004-2005 | CSSE372-02 | Software Project Management | 4 | 14 |
| Winter 2004-2005 | CSSE372-03 | Software Project Management | 4 | 24 |
| Spring 2004-2005 | CSSE376-01 | Software Quality Assurance | 4 | 20 |
| Spring 2004-2005 | CSSE432-01 | Computer Networks | 4 | 24 |
| Spring 2004-2005 | CSSE432-02 | Computer Networks | 4 | 23 |
| Spring 2004-2005 | CSSE491-02 | Model Checking | 4 | 1 |
| Fall 2005-2006 | CSSE373-01 | Formal Methods in Spec & Desn | 4 | 25 |
| Fall 2005-2006 | CSSE497-01 | Senior Project I | 4 | 20 |
| Fall 2005-2006 | CSSE497-02 | Senior Project I | 4 | 8 |
| Winter 2005-2006 | CSSE374-01 | Software Architecture & Design | 4 | 22 |
| Winter 2005-2006 | CSSE498-01 | Senior Project II | 4 | 14 |
| Winter 2005-2006 | CSSE498-02 | Senior Project II | 4 | 16 |
| Spring 2005-2006 | CSSE376-01 | Software Quality Assurance | 4 | 24 |
| Spring 2005-2006 | CSSE497-01 | Senior Project I | 4 | 1 |
| Spring 2005-2006 | CSSE499-01 | Senior Projects III | 4 | 20 |
| Spring 2005-2006 | CSSE499-02 | Senior Projects III | 4 | 9 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 5%**

- Half of that time was spent on development of a WWW History Center.
- The other half was spent advising an inter-university software engineering project.

***Faculty Curriculum Vitae: Donald J. Bagert, Jr.***

**Academic Rank and Title:**

Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August 2002, Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 4 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1986 | Computer Science | Texas A&M University |
| M.S. | 1979 | Computer Science | University of Southwestern Louisiana16 |
| B.S. | 1977 | Engineering | Tulane University |

**Professional Development Activities (AY 2001-2006):**

- Member, ABET Engineering Accreditation Commission, since 2006
- Associate Editor in Chief for Education and Training, IEEE Software, since 2003
- Managing Editor, Forum for Advancing Software engineering Education (FASE) electronic newsletter, 1997-2002 and 2003-04 (still on editorial board)
- Steering Committee Chair, IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), 2000-05 (now Past Chair)
- Chair, IEEE Computer Society Certified Software Development Professional (CSDP) Certification Committee 1 January 2003 to 31 December 2004 (still on committee)
- Member, IEEE Computer Society Educational Activities Board, since 2000
- Member, IEEE Computer Society Software Engineering Program Oversight Committee (SEPOC), since 2005
- Member, Association for Computing Machinery (ACM)
- Member, Institute of Electrical & Electronics Engineers (IEEE) and its Computer Society (IEEE-CS)
- Member, National Society of Professional Engineers (NSPE)
- Member, American Society for Engineering Education (ASEE)
- Attended numerous conferences and workshops in various areas of computer science and software engineering

---

[16] The University of Southwestern Louisiana is now called The University of Louisiana at Lafayette.

- Had published 28 journal articles, conference papers and book chapters

**Professional Experience:**

2002 – Present Professor and Director of Software Engineering
Department of Computer Science and Software Engineering
Rose-Hulman Institute of Technology, Terre Haute, IN

1988 – 2002 Assistant/Associate/full Professor
(Associate Chair 1999-2001)
Department of Computer Science
Texas Tech University, Lubbock, TX

1986 – 1988 Assistant Professor
Department of Computer Science
Northeast Louisiana University17, Monroe LA

1980 – 1986 Instructor
Department of Computer Science
Texas A&M University, College Station, TX

1979 – 1980 Instructor
Department of Computer Science
University of Southwestern Louisiana1, Lafayette, LA

**Consulting Activities:**

- A Detailed Graphical Hierarchy of Electronic Information Desired for a Software Product, project for ASEE-NAVY Summer Faculty Research Program, 1 June to 7 August 1992, Naval Air Warfare Center-Aircraft Division, Warminster, Pennsylvania, USA.
- Member, External Advisory Board, National Science Foundation Division of Undergraduate Education grant DUE #9752482 to the University of Southwestern Louisiana, 1 July 1998-30 June 2000.

**Assigned Duties (AY 2005-2006):**

- Director of Software Engineering – 10 hours per week
- ABET software engineering coordinator – 5 hours per week
- Advisor for 39 students – 5 hours per week

**Institutional Service (AY 2001-2005): None**

**Principal Publications (AY 2001-2006):**

- D. J. Bagert and S. A. Mengel, "Developing and Using a Web-Based Project Process throughout the Software Engineering Curriculum", Journal of Systems and Software, Vol. 74, No. 2, pp. 113-120, 2005.

[17] Northeast Louisiana University is now called The University of Louisiana at Monroe.

- D.J. Bagert and X. Mu, "Current State of Software Engineering Master's Degree Programs in the United States". Proceedings of the Frontiers in Education Conference, Indianapolis, Indiana, USA, 19-22 October 2005, pp. F1G1 to F1G6.
- D.J. Bagert and S. V. Chenoweth, "Future Growth of Software Engineering Baccalaureate Programs in the United States", Proceedings of the ASEE Annual Conference, Portland, Oregon, USA, 12-15 June 2005, CD-ROM, 8 pp.
- D.J. Bagert, "Licensing and Certification of Software Professionals", Advances in Computers, Vol. 60, Academic Press, pp. 1-34, Apr. 2004. Engineering Significance: This is most comprehensive overview of licensing and certification of software professionals to date; covering the topic from an international viewpoint and including several different types of certification mechanisms.
- H. Saiedian, D. J. Bagert, and N. R. Mead, "Software engineering programs: dispelling the myths and misconceptions", IEEE Software, Vol. 19, No. 4, pp. 35-41, Sept. 2002.
- D.J. Bagert, "Education and training in software engineering", Encyclopedia of Software Engineering, Second Edition, John Wiley and Sons, New York, 2002, pp. 452-465.
- E. Fuentetaja and D.J. Bagert, "Software Evolution from a Time-Series Perspective", Proceedings of the International Conference on Software Maintenance, Montreal, Canada, 3-6 October 2002, pp. 226-229.
- D.J. Bagert, M. Barbacci, D. Budgen, T.C. Lethbridge, W. Suryn and H. van Vliet, "Thoughts on Software Engineering Knowledge, and How to Organize It", Proceedings of the Conference on Software Technology and Engineering Practice, Montreal, Canada, 6-8 October 2002, pp. 24-35.

**Selected Honors and Awards:**

- IEEE-CS Golden Core Member, awarded May 2003
- IEEE-CS Outstanding Contribution Award, 2002
- Order of the Engineer, inducted 18 December 1998
- IEEE Senior Member status, since 8 August 1998
- Upsilon Pi Epsilon, inducted 2 December 1981

**Other Scholarly Activity (AY 2001-2006):**

- Professional Registration: Texas, since 1988
- Also an IEEE Computer Society Certified Software Development Professional (roughly the same level as a PE, but for software professionals)

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 24 |
| Fall 2004-2005 | CSSE491-01 | Spanish Web Portal I | 4 | 2 |
| Fall 2004-2005 | CSSE497-01 | Senior Project I | 4 | 12 |
| Winter 2004-2005 | CSSE491-01 | Spanish Web Portal II | 4 | 0 |
| Winter 2004-2005 | CSSE498-01 | Senior Project II | 4 | 12 |

| | | | | |
|---|---|---|---|---|
| Winter 2004-2005 | CSSE499-01 | Senior Projects III | 4 | 2 |
| Spring 2004-2005 | CSSE375-01 | Software Constr & Evolution | 4 | 12 |
| Spring 2004-2005 | CSSE499-01 | Senior Projects III | 4 | 11 |
| Fall 2005-2006 | CSSE371-01 | Software Requirements and Spec | 4 | 25 |
| Fall 2005-2006 | CSSE371-02 | Software Requirements and Spec | 4 | 28 |
| Winter 2005-2006 | CSSE372-01 | Software Project Management | 4 | 37 |
| Winter 2005-2006 | CSSE372-02 | Software Project Management | 4 | 20 |
| Spring 2005-2006 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 12 |
| Spring 2005-2006 | CSSE375-01 | Software Constr & Evolution | 4 | 19 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 20%**

- Research is in software engineering education (involving undergraduate curriculum) and professional issues (licensing and certification of software professionals).

***Faculty Curriculum Vitae: Matthew R. Boutell***

**Academic Rank and Title:**

Assistant Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

Original Appointment Date and Rank: August, 2005, Assistant Professor

Date(s) of Promotion(s) and Resulting Rank(s): N/A

Years of Service at Rose-Hulman: 1 year

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 2005 | Computer Science | University of Rochester |
| M.S. | 2002 | Computer Science | University of Rochester |
| M.Ed. | 1994 | Education | University of Massachusetts |
| B.S. | 1993 | Mathematical Science | Worcester Polytechnic Institute |

**Professional Development Activities (AY 2001-2006):**

- Member, IEEE
- Member, IEEE Computer Society
- Program committee, Workshop on Semantic Learning Applications in Multimedia, IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, 2006.
- Reviewed papers for Pattern Recognition, IEEE Signal Processing Magazine, SPIE Journal of Electronic Imaging.
- Attended 5 major conferences and numerous workshops, presenting at 12 of them.

**Professional Experience:**

2005 – Present Assistant Professor
Computer Science and Software Engineering Department
Rose-Hulman Institute of Technology, Terre Haute, IN

2001 – 2005 Research Intern
Foundation Science Center
Eastman Kodak Company, Rochester, NY

2002 - 2005 Research Assistant
Department of Computer Science
University of Rochester – Rochester, NY

2000 - 2002 Teaching Assistant
Department of Computer Science
University of Rochester – Rochester, NY

1999 - 2000 Adjunct Professor of Computer Science
Department of Computer Science
Stonehill College, Easton, MA

1994 - 2000 Mathematics and Computer Science Teacher
Department of Mathematics
Norton High School, Norton, MA

**Consulting Activities:**

- P&L E-Communications, Rochester, NY – consulted on a prototype video recognition system, 2005.

**Assigned Duties (AY 2005-2006):**

- Advising 13 freshmen with registering for classes for Fall of sophomore year. Total: 8-9 hours of meetings and preparation.

**Institutional Service (AY 2001-2005):**

- Organized bi-weekly Vision Interest Meetings at URCS, 2004-5.
- Graduate Representative to the URCS Faculty, 2003-4.
- Graduate Admissions Committee, URCS, 2002-3.

**Principal Publications (AY 2001-2006):**

- Matthew Boutell, Jiebo Luo, and Robert T. Gray. Sunset scene classification using simulated image recomposition. IEEE International Conference on Multimedia and Expo, Baltimore, MD, July 2003.
- Matthew Boutell and Jiebo Luo. Bayesian fusion of camera metadata cues in semantic scene classification. IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, June 2004.
- Matthew Boutell, Jiebo Luo, and Christopher Brown. Learning spatial configuration models using modified Dirichlet priors. Workshop on Statistical Relational Learning (in conjunction with ICML2004), Banff, Alberta, July 2004.
- Matthew Boutell, Xipeng Shen, Jiebo Luo, and Christopher Brown. Learning multi-label semantic scene classification. Pattern Recognition, 37(9), pp. 1757-1771, September 2004.
- Jiebo Luo and Matthew Boutell. Automatic image orientation detection via confidence-based integration of low-level and semantic cues. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5), pp. 715-726, May 2005.
- Jiebo Luo, Matthew Boutell, Robert T. Gray, and Christopher Brown. Image transform bootstrapping and its applications to semantic scene classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 35(3), June 2005.
- Matthew Boutell, Jiebo Luo, and Christopher Brown. A generalized temporal context model for classifying image collections. ACM Multimedia Systems, 11(1), pp. 82-92, November 2005.

**Selected Honors and Awards:**

- B.S. degree from WPI with High Distinction
- Supported by a federal GAANN fellowship for first three years of PhD studies.
- Award for most papers published by a graduate student, URCS, 2004.
- Best Student Paper award, 2004 IEEE Western New York Image Processing Workshop
- Included three times in Who's Who Among American High School Teachers
- Inducted into Pi Mu Epsilon and Tau Beta Pi honor societies.

- Won Worcester Polytechnic Institute Class of 1879 Prize for Outstanding Project in the Humanities for original composition for brass quintet.

**Other Scholarly Activity (AY 2001-2006, grants, sabbaticals, software development, etc):**

- 3 patent applications filed by Eastman Kodak Company to USPTO.

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2005-2006 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 18 |
| Winter 2005-2006 | CSSE230-01 | Fundamntls of Software Dev III | 4 | 19 |
| Winter 2005-2006 | CSSE230-02 | Fundamntls of Software Dev III | 4 | 14 |
| Spring 2005-2006 | CSSE230-01 | Fundamntls of Software Dev III | 4 | 26 |
| Spring 2005-2006 | CSSE490-01 | Image Recognition | 4 | 16 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 20%**

- Spent serving on a workshop program committee, reviewing papers, writing, submitting, and revising 2 workshop papers and 3 journal papers, and attending 2 international conferences.

***Faculty Curriculum Vitae: Steve Chenoweth***

**Academic Rank and Title:**

Associate Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August, 2003, Associate Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 3 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1990 | Computer Science & Engineering | Wright State University |
| M.S. | 1986 | Computer Science | Wright State University |
| MBA | 1980 | Management | Wright State University |
| M.S. | 1971 | Education | Butler University |
| B.A. | 1966 | Mathematics | Butler University |

**Professional Development Activities in the Last Five Years:**

- Member, IEEE
- Member, IEEE Computer Society
- Member, ACM SIGCSE
- Member, ASEE
- Served as a referee for papers at ASEE Indiana / Illinois regional conference, 2005.
- Co-provider of workshops on teaching software architecture at CSEE&T (Conference on Software Engineering Education & Training, 2004 and 2005.
- Initiated a program to evaluate the effectiveness of the courses of two courses (CSSE 120 and 374), taught in 2004 and 2005.

**Professional Experience:**

| | |
|---|---|
| 1995-2002 | Member of Technical Staff, Bell Laboratories, Lucent Technologies and AT&T, Columbus, OH. |
| 1974-1995 | Consulting Analyst, Senior Systems Analyst, and Manager. NCR Corporation, Dayton, OH. |

**Consulting Activities:**

| | |
|---|---|
| 2002-2003 | Millennium Services, Lincroft, NJ – Software architecture and process consulting. |
| 2005 | Consulted in Indianapolis, for one of the client companies of Rose-Hulman Ventures. |
| 2005 | Worked with Rose-Hulman Ventures, to develop related CSSE classroom activities related to their work. |

**Assigned Duties (AY 2005-2006):**

- Academic advising (15 students), 3 hours per week

**Institutional Service in the last Five Years:**

- Member of committee to create a Masters of Software Engineering program at RHIT.
- One of four members of the Software Engineering curriculum team, 2003-4 and 2004-5.
- Member of Graduate Studies Committee, 2004-5.
- Initiated and directed a monthly department e-newsletter to prospective Rose students.
- Taught a session of RHIT's Catapult program for high school students, 2004 and 2005.
- Academic adviser for up to 20 CS majors during the past two years.

**Principal Publications (AY 2001-2006):**

- "Teamwork and Robotics – A Leadership Experience for Undergraduate Engineering Students," by Steve Chenoweth and David Mutchler. To be presented at ASEE IL-IN, 2006.
- "Methods of Teaching Leadership to Undergraduate Students in Computer Science and Software Engineering," by Steve Chenoweth. Paper and presentation at ASEE IL-IN, March, 2005.
- "Future Growth of Software Engineering Baccalaureate Programs in the United States," by Donald J. Bagert and Stephen V. Chenoweth. Paper was accepted and presented at the national ASEE conference by Dr. Bagert, June, 2005.
- "Project Management: Electrical Engineering vs. Software Engineering," by Steve Chenoweth and Mark A. Yoder. ASEE Peoria Regional Conference, March, 2004.
- "Giving Students Experience in Dealing with Clients In a Course on Requirements," by Donald J. Bagert and Stephen V. Chenoweth. ASEE GSW Regional Conference, March, 2004.

**Other Scholarly Activity (AY 2001-2006):**

- US patent 6,019,394 - Multi-function product label - Assigned to NCR, 1999.
- US patent 5,689,631 - Parallel processing using competing algorithms - Assigned to NCR, 1997.
- Reviewed and consulted on software development projects for Rose Hulman Ventures, for Millennium Services and for Bell Laboratories.

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE371-01 | Software Requirements and Spec | 4 | 13 |
| Fall 2004-2005 | CSSE371-02 | Software Requirements and Spec | 4 | 25 |
| Fall 2004-2005 | CSSE372-03 | Software Requirements and Spec | 4 | 24 |
| Fall 2004-2005 | CSSE413-01 | Artificial Intelligence | 4 | 21 |
| Fall 2004-2005 | CSSE413-02 | Artificial Intelligence | 4 | 22 |
| Fall 2004-2005 | CSSE494-05 | Senior Thesis I | 4 | 1 |

| | | | | |
|---|---|---|---|---|
| Winter 2004-2005 | CSSE374-01 | Software Architecture & Design | 4 | 14 |
| Winter 2004-2005 | CSSE495-05 | Senior Thesis II | 4 | 1 |
| Spring 2004-2005 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 13 |
| Spring 2004-2005 | CSSE490-01 | Software Arch. & Design II | 4 | 8 |
| Spring 2004-2005 | CSSE496-05 | Senior Thesis III | 4 | 0 |
| Fall 2005-2006 | CSSE413-01 | Artificial Intelligence | 4 | 22 |
| Fall 2005-2006 | CSSE413-02 | Artificial Intelligence | 4 | 8 |
| Fall 2005-2006 | CSSE497-03 | Senior Project I | 4 | 16 |
| Fall 2005-2006 | CSSE499-01 | Senior Projects III | 4 | 1 |
| Winter 2005-2006 | CSSE333-01 | Intro to Database Systems | 4 | 28 |
| Winter 2005-2006 | CSSE498-03 | Senior Project II | 4 | 16 |
| Spring 2005-2006 | CSSE377-01 | Software Architecture & Des II | 4 | 22 |
| Spring 2005-2006 | CSSE490-03 | Robotics and Teamwork | 4 | 18 |
| Spring 2005-2006 | CSSE499-03 | Senior Projects III | 4 | 14 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 5%**

- Created two studies of our innovative Teamwork and Robotics course, initial paper published by David Mutchler and myself at ASEE Joint Regional Conference in March.
- Developed a study and program for training incoming freshmen to do course evaluations. This will be delivered in fall, 2006, to 450 freshmen; it includes an experiment, the results of which will be made into a paper on social impacts of students' course evaluations.

***Faculty Curriculum Vitae: Archana Chidanandan***

**Academic Rank and Title:**

Assistant Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August, 2003, Assistant Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 3 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 2004 | Computer Engineering | University of Louisiana at Lafayette |
| Masters | 1999 | Computer Engineering | University of Louisiana at Lafayette |
| B.E. | 1997 | Electronics and Comm. Eng. | Anna University, India |

**Sources of Computer Science Competence:**

**Professional Development Activities (AY 2001-2006):**

- Member, Institute of Electrical and Electronic Engineers (IEEE)
- Member, Association of Computer Machinery(ACM)
- Program committee member, Indiana Women in Computing Conference, InWiC 2006
- Technical program committee member, IEEE Workshop on Signal Processing Systems,2005(SiPS 05)
- Reviewer for IEEE Symposium on Circuits and Systems 2005 (ISCAS 2005), 2004, 2003, American Association for Engineering Education Conference 2005 (ASEE 2005), 10th Asia-Pacific Computer Systems Architecture Conference, 2005, IEEE Conference on Acoustics, Signals and Speech Processing, 2004,2003.
- Evaluator of Student Research Competition submissions at SIGCSE, 2005
- Panel member at a session at the Central Indiana Celebration of Women in Computing Conference, CICWIC 04, Feb. 2004.
- Attended three workshops titled “Active and Co-operative Learning Techniques”, “From Nand to Tetris” and “Facilitating student-written Operating Systems”SIGCSE 2005, St. Louis, MO
- Attended workshop titled “Managing the Academic Career for Faculty Women at Undergraduate Computer Science and Engineering Institutions” – sponsored by Computer Research Association (CRA-W), St. Louis, MO
- Attended workshop titled “Assessment of Student Writing in the Classroom and Across Programs: an Internal and Outcomes Improvement Approach”, Best Assessment Processes VII Symposium, Terre Haute, IN
- Attended seminars titled “Using Linux in Windows Environments”, and “Linux as an alternative desktop”, Illianatech SummIT 2005, Terre Haute, IN
- Attended seminars sponsored by AVID Technolgy, Inc and Mentor Graphics, Indianapolis, IN, May 2005.

- Attended seminars on use of peer-evaluation to assess student work, effective and efficient teaching, use of Dyknow technology in the classroom, place for HSS in the computing and engineering curriculum, best classroom teaching practices – organized by the Quality of Education committee at Rose-Hulman

**Professional Experience:**

2003 – Present Assistant Professor
Computer Science and Software Engineering Department
Rose-Hulman Institute of Technology, Terre Haute, IN

1997 – 1998 Assistant Systems Engineer
Tata Consultancy Services, Madras, India

**Consulting Activities: None.**

**Assigned Duties (AY 2005-2006): None**

**Institutional Service (AY 2001-2005):**

- Member of the Graduate Studies Committee, August 2005 – current
- Member of the Quality of Education Committee, August 2004 – May 2005
- Member of the Visual Arts and Performing Committee, August 2003- May 2004
- Member of the group that worked on the proposal for the Masters in Software Engineering program
- Instructor for Operation Catapult 2005. Worked with students on JAVA programming projects.
- Serve as Academic Advisor to Computer Science and Software Engineering students.
- Served as Freshman Advisor, August 2004-May 2005
- Faculty advisor to a new club called "gEECS" – girls in Electrical Engineering, Computer Science and Software Engineering. The group is applying for club status this year.
- Guided a group of four Computer Science and Computer Engineering female students as they participated in the "Games 4 Girls" programming competition organized by the University of Illinois at Urbana-Champagne. The students were awarded the third prize at the competition.
- Received a CREU grant from CRA-W along with group of two computer engineering students to work an image compression project, 2004-2005.
- Received a CRA-W grant to host a distinguished speaker.
- Hosted a guest speaker from AOL.
- Served as host to a guest speaker at the WWW@10 conference.
- Served on the faculty search committee in the CSSE department, 2004-2005.
- "Introduction to today's Computer Networks", presentation made to the Systems Analysis and Design class at St. Mary of the Woods College, November 2005.

**Principal Publications (AY 2001-2006):**

- A. Chidanandan, J. Moder and M. Bayoumi, "Implementation of NEDA-based DCT architecture using even-odd decomposition of the 8 x 8 DCT matrix," submitted to the Midwest Symposium on Circuits and Systems (MWSCAS 2006).

- A. Chidanandan and M. Bayoumi, “Area-efficient NEDA Architecture for the 1-D DCT/IDCT,” 2006 IEEE International conference on Acoustics, Speech, and Signal Processing (ICASSP06), Toulouse, France
- A. Chidanandan and M. Bayoumi,” Novel Systolic Array Architecture for the Decorrelator using Conjugate Gradient for Least Squares Algorithm,” IEEE Symposium on Circuits and Systems (ISCAS 2005), Kobe, Japan
- Ahmed Shams, Archana Chidanandan, Wendi Pan and Magdy Bayoumi,``NEDA: A new low power high performance DCT architecture,” accepted for publication to the IEEE Transactions in Signal Processing.
- Chidanandan and M. Bayoumi, “Enhanced Parallel Interference Cancellation using Decorrelator for the Base-Station Receiver,” IEEE Symposium in Circuits and Systems (ISCAS 04), Vancouver, Canada
- P. Zhao, P. Golconda, A. Chidanandan, and M. Bayoumi,“A Double Edge Implicit Pulsed Level Convert Flip-Flop for Level Conversion In CVS Systems”, ISVLSI 2004 IEEE CS conference on VLSI 2004, Lafayette, LA
- Shams, W. Pan, A. Chidanandan, M. Bayoumi "A Low Power High Performance Distributed DCT Architecture," ISVLSI 2002 IEEE CS conference on VLSI 2002, Philadelphia, PA

**Other Scholarly Activity (AY 2001-2006): None**

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE232-02 | Computer Architecture I | 4 | 30 |
| Fall 2004-2005 | CSSE232-03 | Computer Architecture I | 4 | 18 |
| Winter 2004-2005 | CSSE232-01 | Computer Architecture I | 4 | 15 |
| Winter 2004-2005 | CSSE332-01 | Operating Systems | 4 | 23 |
| Spring 2004-2005 | CSSE332-01 | Operating Systems | 4 | 11 |
| Spring 2004-2005 | CSSE332-02 | Operating Systems | 4 | 13 |
| Fall 2005-2006 | CSSE232-01 | Computer Architecture I | 4 | 30 |
| Fall 2005-2006 | CSSE232-03 | Computer Architecture I | 4 | 26 |
| Winter 2005-2006 | CSSE332-01 | Operating Systems | 4 | 23 |
| Winter 2005-2006 | CSSE332-02 | Operating Systems | 4 | 23 |
| Spring 2005-2006 | CSSE332-01 | Operating Systems | 4 | 14 |
| Spring 2005-2006 | CSSE432-01 | Computer Networks | 4 | 29 |

***Faculty Curriculum Vitae: Curtis Clifton***

**Academic Rank and Title:**

Assistant Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August, 2005, Assistant Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 1 year |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 2005 | Computer Science | Iowa State University |
| Masters | 2001 | Computer Science | Iowa State University |
| B.S.E.E. | 1992 | Electrical Engineering | Iowa State University |

**Professional Development Activities (AY 2001-2006):**

- Member, Association for Computing Machinery (ACM) – special interest groups in Computer Science Education (SIGCSE), Programming Languages (SIGPLAN), and Software Engineering (SIGSOFT)
- Member, American Society for Engineering Education (ASEE)
- Member Institute of Electric and Electronics Engineers (IEEE) – IEEE Computer Society
- Attended conferences and workshops: OOPSLA, 2000–2005; OOPSLA Educator's Symposium, 2000–2002, 2004; Extravagaria workshop, 2003; AOSD, 2002–2004, 2006; FOAL, 2002–2004, 2006

**Professional Experience:**

| | |
|---|---|
| 2005 – Present | Asst. Professor, Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1998 – 2005 | Research and Teaching Assistant<br>Iowa State University, Ames, IA |
| 1995 – 1998 | Software Team Leader and Sr. Electrical Engineer<br>L&S Electric, Inc., Wausau, WI |
| 1992 – 1995 | Manufacturing Manager<br>Procter & Gamble, Inc., Iowa City, IA |

**Consulting Activities:**

- 1998 – 1999 Independent Consultant Hydroelectric plant automation

**Assigned Duties (AY 2005-2006):**

- Advised rising sophomore during Spring term for Fall registration, about 9 hours total (approx. 15 students, an average of about 20 minutes each; plus around 4 hours of preparation, meeting with their freshmen advisors and studying program requirements)

**Institutional Service (AY 2001-2005):**

- Foundations of Aspect-Oriented Languages (FOAL) Workshop, Co-chair, 2003–2006

- Iowa State Computer Science Graduate Student Organization, President and founder, 1999–2002
- Dept. Chairperson Search Committee, Student representative, 2001
- Reviewer for: Object-Oriented Programs, Systems, Languages, and Applications (OOPSLA), 2004; Trans. on Prog. Lang. and Systems (TOPLAS), 2003; European Conf. on Object-Oriented Prog. (ECOOP), 2003; FOAL, 2004, 2006; Aspect-Oriented Software Development (AOSD), 2003

**Principal Publications (AY 2001-2006):**

- Journal Papers
- 1. Curtis Clifton and Gary T. Leavens. MiniMAO1: An Imperative Core Language for Studying Aspect-Oriented Reasoning. Science of Computer Programming, 2006. To appear.
- 2. Curtis Clifton, Todd Millstein, Gary T. Leavens, and Craig Chambers. MultiJava: Design rationale, compiler implementation, and applications. ACM Trans. on Prog. Lang. and Sys., 2006. To appear. Pre-print available as ISU Tech. Report #04-01b.
- 3. Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the design of JML accommodates both runtime assertion checking and formal verification. Science of Computer Programming, 55(1-3):185–208, March 2005.
- Papers at Refereed Conferences
- 1. Gary T. Leavens and Curtis Clifton. Lessons from the JML Project. In Verified Software: Theories, Tools, Experiments. Zurich, Switzerland. IFIP Working Group 2.3, October 2005.
- 2. Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the design of JML accommodates both runtime assertion checking and formal verification. In Frank S. de Boer, Marcello M. Bonsangue, Susanne Graf, and Willem-Paul de Roever, editors, Formal Methods for Components and Objects: First International Symposium, FMCO 2002, Lieden, The Netherlands, November 2002, Revised Lectures, volume 2852 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2003.
- 3. Curtis Clifton, Gary T. Leavens, Craig Chambers, and Todd Millstein. MultiJava: Modular open classes and symmetric multiple dispatch for Java. In OOPSLA 2000 Conference on Object-Oriented Programming, Systems, Languages, and Applications, volume 35(10) of ACM SIGPLAN Notices, pages 130–145, New York, October 2000. ACM.
- Papers Presented at Refereed Workshops
- 1. Curtis Clifton and Gary T. Leavens. MiniMAO: Investigating the semantics of proceed. In Curtis Clifton, Gary T. Leavens, and Ralf Lämmel, editors, FOAL 2005 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2005, number 05-05 in Computer Science Technical Reports, pages 51–62. Department of Computer Science, Iowa State University, March 2005.
- 2. Curtis Clifton and Gary T. Leavens. Obliviousness, modular reasoning, and the behavioral subtyping analogy. In SPLAT 2003: Software engineering Properties of Languages for Aspect Technologies at AOSD 2003, March 2003. Available as Computer Science Technical Report TR03-01a from ftp//:ftp.cs.iastate.edu/pub/techreports/TR03-01/TR.pdf.
- 3. Curtis Clifton and Gary T. Leavens. Observers and assistants: A proposal for modular aspect-oriented reasoning. In Gary T. Leavens and Ron Cytron, editors, FOAL 2002 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2002, number

02-06 in Computer Science Technical Reports, pages 33–44. Department of Computer Science, Iowa State University, April 2002.

**Selected Honors and Awards:**

- Graduate Scholarship, $900 per semester, Fall 2003, Spring 2004
- Preparing Future Faculty Associate, 2002
- Iowa State University Research Excellence Award, 2001

**Other Scholarly Activity (AY 2001-2006):**

- Co-chaired the Foundations of Aspect-Oriented Languages workshop
- Co-editing the proceeding of the workshop
- Co-authored, with Gary T. Leavens, a $55k NSF grant, "More Modular Reasoning for Aspect-Oriented Programs"
- Implemented the open-source MultiJava Compiler, www.multijava.org
- Participated in implementing the Common Java Modeling Language tools, www.jmlspecs.org

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2005-2006 | CSSE120-03 | Fundamentals of Software Dev I | 4 | 29 |
| Winter 2005-2006 | CSSE333-02 | Intro to Database Systems | 4 | 26 |
| Winter 2005-2006 | CSSE333-03 | Intro to Database Systems | 4 | 18 |
| Spring 2005-2006 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 4 |
| Spring 2005-2006 | CSSE404-01 | Compiler Construction | 4 | 12 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 20%**

- Reviewing papers
- Attending two international conferences
- Writing and submitting papers to two journals and a conference

***Faculty Curriculum Vitae: Lisa C. Kaczmarczyk***

**Academic Rank and Title:**

Assistant Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August, 2005, Assistant Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 1 year |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 2005 | Interdisciplinary | University of Texas at Austin |
| Masters | 1992 | Computer Science | University of Oregon |
| Masters | 1990 | Information Systems | Northeastern University |
| B.S. | 1983 | Drama, Spanish | Tufts University |

**Professional Development Activities (AY 2001-2006):**

- Member, Cognitive Science Society
- Member, Association of Computing Machinery (Special Interest Group on Computer Science Education)
- Member, IEEE Computer Society
- Attended Frontiers in Education Conference, Indianapolis, IN., 2005.
- Coordinator (co) of Student Volunteers, SIGCSE Conference, 2006, Houston, TX.
- Attended Computing Research Association Academic Careers Workshop, Washington DC, 2006.
- Attended first International Computing Education Research Workshop (ICER), 2005, Seattle, WA.

**Professional Experience:**

| | |
|---|---|
| 1999 – 2005 | Assistant Instructor<br>Department of Computer Sciences<br>University of Texas at Austin, Austin, TX |
| Summer 2001 | Visiting Lecturer<br>University of Oregon, Eugene, OR |
| 1992 – 1999 | Instructor<br>Chemeketa Community College, Salem, OR |
| 1999 – 2000 | Software developer<br>Pointserve Inc, Austin, TX |

**Consulting Activities: None.**

**Assigned Duties (AY 2005-2006): None**

**Institutional Service (AY 2001-2005):**

- Reviewer for the National Science Foundation – Multiple occasions

- Invited Talk: "The Acquisition of Intellectual Expertise: A Computational Theory." (2003, January) School of Computing, University of Kent, Canterbury, UK.
- Invited Talk: "Neural Networks and Modeling of Student Learning." (2001, June) School of Computing, University of Leeds, UK.
- Invited Article: Kaczmarczyk, L. C. (2001b, December). "Curriculum descant: Is AI abstract and impractical? Isn't the answer obvious?" Intelligence, 12, 19-20.
- Kaczmarczyk (Moderator), Kruse, Lopez, Kumar (2004) "Incorporating Writing into the CS Curriculum." Proceedings of the 35th Annual SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2004), Norfolk, Virginia.

**Principal Publications (AY 2001-2006):**

- Kaczmarczyk, L. C., Last, M. Z, Miikkulainen, R. (in press). "The Effect of Delivery Method on Conceptual and Strategy Development". To appear in Proceedings of the 28th Annual Conference of the Cognitive Science Society, Vancouver, British Columbia.
- Kaczmarczyk, L.C., Miikkulainen, R. (2004). "The Acquisition of Intellectual Expertise: A Computational Model". Proceedings of the 26th Annual Conference of the Cognitive Science Society, Chicago, Illinois.
- Kaczmarczyk, L. C. (2003). "A Technical Writing Class for Computer Science Majors: Measuring Student Perceptions of Learning." Proceedings of the 34th Annual Technical Symposium on Computer Science Education (SIGCSE 2003), Reno, Nevada.
- Kaczmarczyk, L. C. (2001a, June). "Accreditation and student assessment: why we all need to pay attention." Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2001), Canterbury, UK.

**Selected Honors and Awards:**

- Invited Keynote Speaker, Indiana Women in Computing Conference, 2006
- Awarded scholarship to The Grace Hopper Celebration of Women in Computing, 2004
- Awarded scholarship to The Grace Hopper Celebration of Women in Computing, 2002

**Other Scholarly Activity (AY 2001-2006): None**

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2005-2006 | CSSE221-02 | Fund. of Software Dev Honors | 4 | 26 |
| Winter 2005-2006 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 23 |
| Winter 2005-2006 | CSSE120-02 | Fundamentals of Software Dev I | 4 | 26 |
| Spring 2005-2006 | CSSE120-02 | Fundamentals of Software Dev I | 4 | 6 |
| Spring 2005-2006 | CSSE490-02 | Great Papers in CS & SE | 4 | 9 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 15%**

- Revised and submitted research paper for submission to cognitive science conference (see "in press" publication)
- Worked on poster development for poster to be presented at Cognitive Science conference – goes with the accepted paper

- Worked on grant proposal for submission to NSF in collaboration with co-PIs from two other universities (submitted; outcome August 2006)
- Worked on development of INWiC keynote speech; this included conducting multiple informal interviews with women students
- Investigated ideas for future research directions after attending the CRA Academic Careers Workshop
- Met repeatedly with several undergraduate students to develop plans for conducting a senior research thesis (two of these currently scheduled to do thesis with me next year)

***Faculty Curriculum Vitae: J.P. Mellor***

**Academic Rank and Title:**

Associate Professor of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | November, 1999, Assistant Professor. |
| Date(s) of Promotion(s) and Resulting Rank(s): | September, 2005, Associate Professor. |
| Years of Service at Rose-Hulman: | 7 years |

**Degrees:**

| | | |
|---|---|---|
| Ph.D. | 2000 | Electrical Engineering and Computer Science<br>Massachusetts Institute of Technology. |
| S.M. | 1995 | Electrical Engineering and Computer Science<br>Massachusetts Institute of Technology. |
| B.S. | 1990 | Nuclear Technology<br>University of the State of New York. |
| B.S. | 1987 | Electrical Engineering<br>United States Naval Academy. |

**Professional Development Activities (AY 2001-2006):**

- Membership in Scientific and Professional Societies: Institute of Electrical and Electronics Engineers (IEEE) Association for Computing Machinery (ACM) Sigma Xi, The Scientific Research Society Tau Beta Pi
- Created several new courses including Computer Vision and Advanced Operating Systems.
- Initiated major revisions to the Computer Architecture sequence.
- Published articles (and presentations) in the area of computer science.
- Participated in several research grants.
- Assisted several clients of Rose-Hulman Ventures.

**Professional Experience:**

| | |
|---|---|
| 2002--Present | Director, Imaging Systems Laboratory<br>Rose-Hulman Institute of Technology, Terre Haute, IN. |
| 1999--Present | Asst. Professor/Assoc. Professor<br>Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN. |
| 2001--2005 | Innovation Fellow Project Manager<br>Rose-Hulman Ventures, Terre Haute, IN. |
| 1997--1999 | System Administrator<br>Zoesis Studios, Newton, MA. |
| 1996--1999 | Custom Software Developer<br>Advanced Unix System Administration Instructor<br>Great Eastern Technology, Woburn, MA. |

1992--1993 Material Assistant
Submarine Squadron Three, San Diego, CA.

1990--1992 Main Propulsion Assistant
U.S.S. Drum, SSN-677, San Diego, CA.

1989--1990 Shift Engineer/Assistant Shift Engineer
Naval Nuclear Power Training Unit, Ballston Spa, NY.

**Consulting Activities:**

- 2000--2001 Expert Witness
  Kirkland Ellis Law Firm, Chicago, IL.

**Assigned Duties (AY 2005-2006): None (sabbatical)**

**Institutional Service (AY 2001-2005):**

- Imaging Systems Certificate Advisor, 2002--present.
- Program committee member for the WWW@10: The Dream and the Reality conference, 2004.
- Faculty advisor for the Rose-Hulman Computing Society.
- Faculty teller, 2002-2004.
- Laptop Committee 2000-2005
- Parallel Computing Steering Group 2000-Present

**Principal Publications (AY 2001-2006):**

- "Preliminary Findings on the Clinical Utility of an Unobtrusive Home Health Monitoring System," P.A. Woodbridge, J.P. Mellor and M. Weiner, Human-Computer Interaction International, July 2005.
- "An Improved Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, Workshop on Applications of Computer Vision, January 2005.
- "An Improved Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, Sandor Pethes, Joshua Starr, Kevin Gorsky, and Charles Lehman, Technical Report RHV-TR-2, Rose-Hulman Institute of Technology, Rose-Hulman Ventures, September 2003.
- "An Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, British Machine Vision Conference, September 2003.
- "New Approach to Creating a Successful Technology-Based Incubator," Brij Khorana, John Davidson, J.P. Mellor, National Business Incubator Association, May 2003.
- "Geometry and Texture from Thousands of Images," J.P. Mellor, International Journal of Computer Vision, 51(1):5--35, January 2003.
- "Engineering Education For Inclusive Design," Lawrence Goldberg, Eric Jolly, J.P. Mellor, Babette Moeller, Madeleine Rothberg, Richard Stamper, Michael Wollowski, Frontiers in Education, November 2002.
- "An Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, Mike Oder, Joshua Starr, and Joel Meador, Technical Report RHV-TR-1, Rose-Hulman Institute of Technology, Rose-Hulman Ventures, April 2002.

**Other Scholarly Activity (AY 2001-2006):**

- Patent:  System and Method for Transforming Graphical Images..
- Certified as Chief Engineer of Naval Nuclear Propulsion Plants.
- Certified for supervision, operation, and maintenance of Naval Nuclear Propulsion Plants.

**Courses taught (AY 2003-2005):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2003-2004 | CSSE220-04 | Fundamntls of Software Dev II | 4 | 23 |
| Fall 2003-2004 | CSSE491-01 | Directed Independent Studies | 4 | 1 |
| Fall 2003-2004 | CSSE494-01 | Senior Thesis I | 4 | 2 |
| Winter 2003-2004 | CSSE490-01 | Adv Operating Systems and UNIX | 4 | 24 |
| Winter 2003-2004 | CSSE490-02 | Adv Operating Systems and UNIX | 4 | 8 |
| Winter 2003-2004 | CSSE495-01 | Senior Thesis II | 4 | 2 |
| Spring 2003-2004 | CSSE461-01 | Computer Vision | 4 | 19 |
| Fall 2004-2005 | CSSE232-01 | Computer Architecture I | 4 | 20 |
| Fall 2004-2005 | CSSE494-06 | Senior Thesis I | 4 | 1 |
| Winter 2004-2005 | CSSE220-02 | Fundamntls of Software Dev II | 4 | 25 |
| Winter 2004-2005 | CSSE490-01 | Ad Oper Systems & UNIX | 4 | 10 |
| Winter 2004-2005 | CSSE495-06 | Senior Thesis II | 4 | 1 |
| Spring 2004-2005 | CSSE461-01 | Computer Vision | 4 | 13 |
| Spring 2004-2005 | CSSE496-06 | Senior Thesis III | 4 | 1 |
| Winter 2005-2006 | CSSE491-01 | Advanced Operating Systems | 4 | 1 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 50%**

- Working with Elastic Image to further develop and bring to market technology which enables thermoformed plastic with significant three-dimensional shape to be cost effectively manufactured with high quality decoration.

***Faculty Curriculum Vitae: Laurence D. Merkle***

**Academic Rank and Title:**

Assistant Professor of Computer Science and Software Engineering (on tenure track)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | September, 2002, Assistant Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | N/A |
| Years of Service at Rose-Hulman: | 4 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1996 | Computer Engineering | Air Force Institute of Technology |
| M.S.C.E. | 1992 | Computer Engineering | Air Force Institute of Technology |
| B.S. | 1987 | Comp. & Sys. Engineering | Rensselaer Polytechnic Institute |

**Sources of Computer Science Competence:**

- Graduate coursework subsumed that required for computer science programs
- Graduate and subsequent research has been in areas relevant to computer science (including evolutionary computation, parallel computation, computational science and engineering, and information assurance)
- Nine years of experience teaching computer science at the undergraduate level – 20 distinct courses at four different institutions

**Professional Development Activities (AY 2001-2006):**

- Professional societies: ACM, IEEE (Senior Member), ASEE, AAAI
- Visiting Professor, Information Institute, Air Force Research Laboratory.
- Information Warfare Applications Course
- Program Manager for High Performance Computing, Air Force Office of Scientific Research.

**Professional Experience:**

| | |
|---|---|
| 2002 – Present | Assistant Professor<br>Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1999 – 2002 | Assistant/Associate Professor<br>Computer Science Department<br>United States Air Force Academy, Colorado Springs, CO |
| 1997 – 1998 | Adjunct Professor<br>Chapman University and College of Santa Fe, Albuquerque, NM |
| 1996 – 1999 | Chief, Center for Plasma Theory and Computation<br>Air Force Research Laboratory, Albuquerque, NM |
| 1988 – 1991 | Artificial Intelligence Project Officer<br>Air Force Materiel Command, Dayton, OH |

**Consulting Activities: None.**

**Assigned Duties (AY 2005-2006):**

- Faculty Affairs Committee (1 hour per week)
- Greek Advisory Council (1 hour per term)

**Institutional Service (AY 2001-2005):**

- Committees: Faculty Affairs, Rules and Discipline, Quality of Education, Greek Advisory Council
- M.S. Thesis Committee Member (Dillman, Morgan, Schmitt, Yargaladda)
- Faculty Advisor (Upsilon Pi Epsilon, Tau Beta Pi, Pi Kappa Alpha, Rose Chorus)

**Principal Publications (AY 2001-2006):**

- "Underwater Hacker Missile Wars: A Cryptography and Engineering Contest," Joshua Holden, Richard Layton, Laurence Merkle, and Tina Hudson, Cryptologia, Vol. 30, (pp. 69-77), 2006.
- "Multi-Robot Cooperation Using the Ant Algorithm with Variable Pheromone Placement," Eric Borzello and Laurence Merkle, proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005.
- "Underwater Model Rockets: An Innovative Design Problem and Competition for Undergraduate Students in Engineering, Math and Science," Richard Layton, Josh Holden, Tina Hudson, and Laurence Merkle, proceedings of the ASEE Annual Conference, Session #3566, Portland, Oregon, June 12-15, 2005. Best paper in Mechanical Engineering Division.
- "Automated Load Balancing of a Missile Defense Simulation Using Domain Knowledge," Martin C. Carlisle and Laurence D. Merkle, Journal of Defense Modeling and Simulation, Vol. 1, No. 1, (pp. 59-68), April 2004.
- "Design Optimization for a Novel Class of High Power Microwave Sources," Laurence D. Merkle and John W. Luginsland, invited paper for the IEEE Congress on Evolutionary Computation special session on Evolutionary Design Optimization, Canberra, Australia, 2003.
- "Measuring the Effectiveness of Robots in Teaching Computer Science," Barry Fagin and Laurence Merkle, proceedings of the Technical Symposium on Computer Science Education, Reno, Nevada, 2003.
- "Towards Effective Evolutionary Algorithms for Polypeptide Structure Prediction," Gary B. Lamont and Laurence D. Merkle, in "Evolutionary Computation in Bioinformatics, " ed. Gary Fogel and David W. Corne, 2002.
- "Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education," Barry S. Fagin and Laurence Merkle, ACM Journal of Educational Resources in Computing, Vol. 2, No. 4, (pp. 1-18), December, 2002.

**Selected Honors and Awards:**

- Best Paper, Mechanical Engineering Division, 2005 American Society for Engineering Education Annual Conference & Exposition
- USAFA Department of Computer Science Research Excellence Award, 2001-2002
- Upsilon Pi Epsilon, Eta Kappa Nu, Tau Beta Pi

**Other Scholarly Activity (AY 2001-2006):**

- Sponsored research: Evolutionary Computation in Polymorphous Computing Architectures, $64K, Advanced Computing Architectures Branch, Air Force Research Laboratory.
- Program Committee (Genetic and Evolutionary Computation Conference (GECCO); Technical Symposium on Computer Science Education; ACM Symposium on Applied Computing; IEEE International Conference on Systems, Man and Cybernetics; IEEE Information Assurance Workshop)
- Chair/co-chair, Undergraduate Workshop, GECCO
- Co-chair, Military and Security Applications of Evolutionary Computation, GECCO
- Development of Web-based SIGCSE Conference Registration System

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE494-03 | Senior Thesis I | 4 | 2 |
| Winter 2004-2005 | CSSE332-02 | Operating Systems | 4 | 21 |
| Winter 2004-2005 | CSSE332-03 | Operating Systems | 4 | 23 |
| Winter 2004-2005 | CSSE495-03 | Senior Thesis II | 4 | 2 |
| Spring 2004-2005 | CSSE442-01 | Computer Security | 4 | 22 |
| Spring 2004-2005 | CSSE442-02 | Computer Security | 4 | 19 |
| Spring 2004-2005 | CSSE491-01 | Algorithms for Classrm Schdlng | 4 | 1 |
| Spring 2004-2005 | CSSE496-03 | Senior Thesis III | 4 | 2 |
| Fall 2005-2006 | CSSE232-02 | Computer Architecture I | 4 | 25 |
| Fall 2005-2006 | CSSE494-04 | Senior Thesis I | 4 | 2 |
| Winter 2005-2006 | CSSE232-01 | Computer Architecture I | 4 | 20 |
| Winter 2005-2006 | CSSE495-04 | Senior Thesis II | 4 | 2 |
| Spring 2005-2006 | CSSE332-02 | Operating Systems | 4 | 21 |
| Spring 2005-2006 | CSSE442-01 | Computer Security | 4 | 21 |
| Spring 2005-2006 | CSSE496-04 | Senior Thesis III | 4 | 2 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 20%**

- Approximately 500 hours spent developing Web-based SIGCSE Conference Registration System
- Approximately 400 hours spent researching EA-based generation of compiler heuristics for polymorphous computing architectures
- Attended Technical Symposium on Computer Science Education
- Attended and organized workshops for Genetic and Evolutionary Computation Conference
- Prepared paper for publication at (and attended) Congress on Evolutionary Computation
- Attended IEEE Information Assurance Workshop

***Faculty Curriculum Vitae: David Mutchler***

**Academic Rank and Title:**

Professor of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | September, 1994, Associate Professor |
| Date(s) of Promotion(s) and Resulting Rank(s): | September, 2000, Full Professor |
| Years of Service at Rose-Hulman: | 12 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph.D. | 1986 | Computer Science | Duke University |
| M.S. | 1980 | Mathematics | University of Virginia |
| B.A. | 1976 | Mathematics | University of Virginia |

**Professional Development Activities (AY 2001-2006):**

- Membership in Scientific and Professional Societies: Association of Computing Machinery (ACM). American Association for Artificial Intelligence (AAAI). Fulbright Academy . Special Interest Group for Computer Science Education (SIGCSE). Special Interest Group for Artificial Intelligence (SIGART)
- Conferences attended include:
- National Botball Robotics Education Conference, Jacksonville, Florida, July 14-17, 2005.
- The First Annual Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-2005), Marina del Rey, California, June 1-3, 2005
- Hewlett-Packard University Mobile Technology Solutions Conference 2004, San Jose, CA, Nov. 4-5, 2004.
- "The World Wide Web at 10: The Dream and the Reality," Rose-Hulman Institute of Technology, Oct. 2004.
- Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, California, July 25-29, 2004.
- 2004 American Association for Artificial Intelligence (AAAI) Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education, at Stanford University, March 24-26, 2004.
- Hewlett-Packard University Mobile Technology Solutions Conference 2003, Palo Alto, October 16-17, 2003.
- Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 9-15, 2003.
- Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, July 28 – Aug. 1, 2002.
- 3rd International Conference on Computers and Games, Edmonton, Alberta, Canada, July 25 - 27, 2002.
- National Botball Robotics Education Conference, Norman, Oklahoma, June 29 - July 2, 2002.

- Robotics Educators Workshop, Norman, Oklahoma, January 10-13, 2002.
- Talks not associated with publications listed above include:
- "Using Tablet PCs creatively," Best Assessment Processes VI Symposium, Rose-Hulman, February, 2004.
- "Teaching (Technically) by Storytelling" at Rose-Hulman's Opening Symposium, August 2002.
- Sabbatical:
- Fulbright Award for Lecturing/Research, University of Mauritius, Mauritius, 2000-2001.

**Professional Experience:**

| | |
|---|---|
| 1994 – Present | Assoc. Professor/Professor<br>Department of Computer Science and Software Engineering<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1987 – 1994 | Assistant Professor<br>University of Tennessee, Knoxville, TN |
| 1986 – 1987 | Research Computer Scientist<br>Naval Research Laboratory, Washington, DC |
| 1981 – 1986 | Research Assistant<br>Duke University, Durham, NC |
| 1980 – 1981 | Instructor<br>Davidson College, Davidson, NC |
| 1976 – 1980 | Teaching Assistant<br>University of Virginia, Charlottesville, VA |

**Consulting Activities: None.**

**Assigned Duties (AY 2005-2006):**

- Laptop committee (0.5 hours per week)
- Academic Computing committee (0.5 hours per week)
- Informal liaison for Tablet PC program (1.0 hours per week)
- Faculty advisor for Unity and Board Games clubs (0.1 hours per week)

**Institutional Service (AY 2001-2005):**

- Coordinator of the Midwest Regional Botball program, 1997 to date.
- Leader of various area First Lego League teams, 1998 to date.
- Co-coordinator of the Tablet PC and DyKnow program at Rose-Hulman, 2002 to date.
- Member of planning and program committees for "The World Wide Web at 10: The Dream and the Reality," Rose-Hulman. October 2004.
- Faculty advisor for Unity, a Rose-Hulman club whose mission is "to provide a place where individuals can be themselves without fear of prejudice or hate, and to promote an environment on campus and throughout our community that is welcoming to all people," 2001 to date.
- Faculty advisor for the Board Games club, 1995 to date.

- Suggested a "Safe Zone" project that the Diversity Council sponsored, implemented during 2004-2005.
- Frequent leader of activities for Explore Engineering outreach program.
- Member of many committees at Rose-Hulman.

**Principal Publications (AY 2001-2006):**

- Joint with colleagues from Rose-Hulman, presented at the First Workshop on the Impact of Pen-based Technology on Education, Purdue University, April 6-7, 2006:
- "Tablet or Laptop? Does the pen matter, and if so, in what way?"
- "An interdisciplinary project using handheld, pen-based devices for project management, technical writing for mobile devices, and software development."
- "Pen-based tablet PCs with DyKnow in introductory 'studio' physics classes."
- "Implementing Tablet Computers in the technical communication classroom: measuring the impact of mobility on communication skills."
- Co-PI of $100,000 funded grant from Microsoft, "Assessing and Evaluating the Symbiosis of Tablet PCs and Collaboration-Facilitating Software in the Classroom," February 2006.
- Joint with colleagues from Rose-Hulman: "The world is our classroom: Tablet PCs", Hewlett-Packard University Mobile Technology Solutions Conference 2004, San Jose, California, November 4-5, 2004.
- Co-PI of $220,000 and $135,500 funded grants from Hewlett-Packard, "Using Mobile Technology to Amplify Interactions and Enhance Learning in the Classroom and Laboratory," February 2003 and February 2004.
- "The role of Botball teachers: a teacher-as-facilitator model," and "How and why to run a Botball 'Midway' Tournament," National Botball Robotics Education Conference, Norman, OK, June 29 - July 2, 2002.
- Lynn Kiaer, David Mutchler and Jeffrey Froyd, "Laptop computers in an integrated first-year curriculum," Communications of the ACM 41(1), pp. 45-49 (January, 1998).

**Selected Honors and Awards:**

- Fulbright Award for Lecturing/Research, University of Mauritius, Mauritius, 2000-2001.
- Outstanding Faculty Member of the Week, as selected by President Samuel F. Hulbert, Rose-Hulman Institute of Technology: Many times.
- 1992-93 ACM Teacher of the Year, as selected by the Univ. of Tennessee chapter of the ACM.
- 1989-90 ACM Teacher of the Year, as selected by the Univ. of Tennessee chapter of the ACM.

**Other Scholarly Activity (AY 2001-2006):**

- Sabbatical at the University of Mauritius, per a Fulbright Award for Teaching and Research, 2000-2001

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE120-02 | Fundamentals of Software Dev I | 4 | 24 |
| Fall 2004-2005 | CSSE120-03 | Fundamentals of Software Dev I | 4 | 29 |
| Winter 2004-2005 | CSSE120-01 | Fundamentals of Software Dev I | 4 | 22 |
| Winter 2004-2005 | CSSE120-03 | Fundamentals of Software Dev I | 4 | 17 |
| Spring 2004-2005 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 27 |
| Spring 2004-2005 | CSSE479-01 | Cryptography | 4 | 4 |
| Spring 2004-2005 | CSSE490-03 | Robotics and Teamwork | 4 | 23 |
| Spring 2004-2005 | CSSE491-03 | Advanced CGI Programming | 4 | 1 |
| Fall 2005-2006 | CSSE120-02 | Fundamentals of Software Dev I | 4 | 18 |
| Fall 2005-2006 | CSSE221-01 | Fund. of Software Dev Honors | 4 | 16 |
| Fall 2005-2006 | CSSE494-03 | Senior Thesis I | 4 | 2 |
| Winter 2005-2006 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 11 |
| Winter 2005-2006 | CSSE220-02 | Fundamntls of Software Dev II | 4 | 20 |
| Winter 2005-2006 | CSSE495-03 | Senior Thesis II | 4 | 2 |
| Spring 2005-2006 | CSSE332-03 | Operating Systems | 4 | 23 |
| Spring 2005-2006 | CSSE490-04 | Robotics and Teamwork | 4 | 16 |
| Spring 2005-2006 | CSSE491-01 | Directed Independent Studies | 4 | 1 |
| Spring 2005-2006 | CSSE496-03 | Senior Thesis III | 4 | 2 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 10%**

- 5 weeks of 50 weeks, as follows: Attend AIIDE conference (1 week); Attend National Conference on Robotics Education (1 week); write articles for and attend WIPTE conference (2 day conference, 3 days writing); Research in using Tablet PCs/DyKnow (1 week spread over the year); Work with my senior thesis students (1 week spread over the year).

***Faculty Curriculum Vitae: Michael Wollowski***

**Academic Rank and Title:**

Associate Professor of Computer Science and Software Engineering (with tenure)

**Rose-Hulman Service:**

| | |
|---|---|
| Original Appointment Date and Rank: | August 1999, Assistant Professor |
| Date(s) of promotion(s) and Resulting Rank(s) | August 2005, Associate Professor |
| Years of Service at Rose-Hulman: | 7 years |

**Degrees:**

| | | | |
|---|---|---|---|
| Ph..D. | 1998 | Computer Science | Indiana University, USA |
| M.S. | 1988 | Computer Science | Indiana University, USA |
| Vordiplom, | 1986 | Informatik | Universitat Hamburg, Germany |

**Professional Development Activities in the Last Five Years:**

- Member, Association of Computing Machinery
- Member, Web Intelligence Consortium
- Reviewer for ACM's Computing reviews
- Chaired sessions at several conferences
- Attended the conferences listed under Publications
- Attended many of the Teaching Enhancement seminars given over the years here at RHIT
- Attended the 2002 Search Engine Meeting in San Francisco, CA, from April 15-16.
- Attended The Tenth Annual Teaching and Learning with Technology Conference, West Lafayette, IN, March 1 - 2, 2006.
- Attended the 2005 Frontiers in Education Conference, Indianapolis, IN, October 19-22, 2005.
- Attended The Ninth Annual Teaching and Learning with Technology Conference, West Lafayette, IN, March 1 - 2, 2005.
- Attended the Informatics: Defining the Research Agenda Conference, Bloomington, IN, Sep 10 – 12, 2005.
- Robert Signorelli and myself gave a presentation of our work entitled Towards a General-Purpose Search Engine at the 2003 Search Engine Meeting, Boston, April 2003.

**Professional Experience:**

| | |
|---|---|
| 1999 – Present | Asst./Assoc. Professor<br>Computer Science and Software Engineering Department<br>Rose-Hulman Institute of Technology, Terre Haute, IN |
| 1995 – 1999 | Visiting Assistant Professor<br>Computer Science Department, Siena College, Loudonville, NY. |
| 1990 – 1995 | Associate Instructor<br>Computer Science Department, Indiana University, Bloomington, IN. |

**Consulting Activities: None**

**Assigned Duties (AY 2005-2006):**

- Member of the "Advisory Committee for Computer Use", met for about 10 hours during the fall quarter.
- Chair of "Rules and Discipline Committee", 1 hour/week.
- Advising: 6 hours/term

**Institutional Service (AY 2001-2005):**

- Board member of Engenius Solutions.
- Faculty advisor to the Web-based Soda Machine project
- Served on several committees: Quality of Education, International Programs, and Rules and Discipline (chair), Advisory Committee of Computer Use
- Actively participated in the hiring of all of our new colleagues.
- Developed several new courses: CSSE 120, CSSE 220, CSSE 481, CSSE 490: Swarm Intelligence. Some of those courses were developed together with Andy Kinley and David Mutchler. Revised CSSE 230.
- Pioneered the use of the web in our courses.
- Academic advisor to about 20 students
- Gave several workshops to students and faculty about creating web pages
- Chaired the Technology Enhanced Instruction Team, recommended directions that Rose-Hulman might employ and support in the areas of web-based instruction, technology in the classroom, and in general any other technology that might enhance our ability to provide the best learning experience possible.

**Principal Publications (AY 2001-2006):**

- Michael Wollowski. A Theorem Prover for a Diagrammatic Blocks World. Proceedings of The 2006 Midwest Artificial Intelligence and Cognitive Science Conference, Valparaiso, IN, April 1-2, 2006.
- Michael Wollowski. Search and Inference with Diagrams. Proceedings of The Ninth IASTED International Conference on INTERNET & MULTIMEDIA SYSTEMS & APPLICATIONS, Honolulu, HI, Aug 15 – 17, 2005.
- Michael Wollowski. Living in a Transparent Future: Search in a Wired World. Online Proceedings of the WWW@10 Conference, Terre Haute, IN, Sep 30 – Oct 2, 2004.
- Michael Wollowski, Peter Nei, Chris Barrell. A Diagrammatic Inference System for the Web. Proceedings of The Thirteenth International World Wide Web Conference, Alternate Track Papers & Posters, pp. 374-375. NY, NY, 2004.
- Michael Wollowski, An XML-Based Syllabus Editor and Search Engine. Proceedings of the Stop Surfing - Start Teaching 2003 National Conference, Las Vegas, 2003, pp 107 -111.
- Michael Wollowski, XML Based Course Websites. Proceedings of the E-Learn 2002 Conference, Montreal, 2002, pp 1043 - 1048.
- Michael Wollowski, An Undergraduate Research Course Aimed at Furthering the Web, Proceedings of the 2001 Frontiers in Education Conference, Reno, NV, from Oct 10 - 13.

**Other Scholarly Activity (AY 2001-2006, grants, sabbaticals, software development, etc):**

- Defensive disclosure: Michael Wollowski, Robert Signorelli, Chris Barrell: A Method and System for User Initiated Repeat Purchases via the Internet on the IP.com Prior Art Database.

**Courses taught (AY 2004-2006):**

| TERM | COURSE | DESCRIPTION | CREDIT | ENRL |
|---|---|---|---|---|
| Fall 2004-2005 | CSSE221-01 | Fund. of Software Dev Honors | 4 | 24 |
| Fall 2004-2005 | CSSE221-02 | Fund. of Software Dev Honors | 4 | 15 |
| Fall 2004-2005 | CSSE494-04 | Senior Thesis I | 4 | 2 |
| Winter 2004-2005 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 15 |
| Winter 2004-2005 | CSSE481-01 | Web-Based Information Systems | 4 | 10 |
| Winter 2004-2005 | CSSE490-02 | Swarm Intelligence | 4 | 24 |
| Winter 2004-2005 | CSSE495-04 | Senior Thesis II | 4 | 1 |
| Spring 2004-2005 | CSSE304-01 | Programming Language Concepts | 4 | 27 |
| Spring 2004-2005 | CSSE304-02 | Programming Language Concepts | 4 | 27 |
| Spring 2004-2005 | CSSE496-04 | Senior Thesis III | 4 | 1 |
| Fall 2005-2006 | CSSE220-01 | Fundamntls of Software Dev II | 4 | 13 |
| Fall 2005-2006 | CSSE481-01 | Web-Based Information Systems | 4 | 28 |
| Fall 2005-2006 | CSSE494-01 | Senior Thesis I | 4 | 2 |
| Winter 2005-2006 | CSSE474-01 | Theory of Computation | 4 | 20 |
| Winter 2005-2006 | CSSE474-02 | Theory of Computation | 4 | 28 |
| Winter 2005-2006 | CSSE490-01 | Swarm Intelligence | 4 | 22 |
| Winter 2005-2006 | CSSE491-02 | Directed Independent Studies | 4 | 2 |
| Winter 2005-2006 | CSSE491-03 | Advanced MMD Networking | 4 | 1 |
| Winter 2005-2006 | CSSE495-01 | Senior Thesis II | 4 | 2 |
| Spring 2005-2006 | CSSE304-01 | Programming Language Concepts | 4 | 21 |
| Spring 2005-2006 | CSSE304-02 | Programming Language Concepts | 4 | 19 |
| Spring 2005-2006 | CSSE304-03 | Programming Language Concepts | 4 | 12 |
| Spring 2005-2006 | CSSE491-02 | Advanced MMO Networking | 4 | 1 |
| Spring 2005-2006 | CSSE496-01 | Senior Thesis III | 4 | 2 |

**Percentage of Time Devoted to Scholarly Activity (AY 2005-2006): 10%**

- Reviewed literature in Swarm Intelligence and Web-based Information Systems.
- Prepared a paper for publication and presentation.

## 4. Faculty Size and Workload

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> There are enough full time faculty members to provide continuity, oversight and stability, to cover the curriculum reasonably, and to allow an appropriate mix of teaching, professional development, scholarly activities and service for each faculty member. The faculty assigned to the program has appropriate authority for the creation, delivery, evaluation and modification of the program, and the responsibility for the consistency and quality of its courses.

### A. Faculty Size

All required courses and all department electives are offered annually. There are no difficulties in meeting this frequency of offering.

### B. Faculty Workload

The institute has a standard that all faculty are entitled to 20% of their time for professional work. The department works on a 12-contact hour teaching load, which means faculty teach either three sections of courses that meet four hours per week (lecture courses) or two sections of courses that meet six hours per week (laboratory courses). Most faculty have one day per week that they do not teach any classes, and some faculty have two days per week of no teaching requirement. Faculty have either one or two preparations per term.

Advising of students is distributed relatively evenly over all faculty in the department, insuring each faculty member does not have an unduly large number of advisees. Freshmen students are assigned to faculty who volunteer to take on the extra attention these students need in adjusting to college life. Several faculty members in the department volunteer for this program. Upperclass students are assigned to a faculty member by academic year, and stay with that advisor until they graduate. Student advising is considered part of a faculty member's service obligation to the department and the school, and is taken into consideration during annual performance reviews, as well as at promotion and tenure time.

The department head is granted 2/3 release time for administrative duties. One member of the department is appointed as the Director of Software Engineering and is granted 1/3 release time to oversee that program.

### C. Program Development and Delivery

The department faculty regularly gather information from the department's Board of Advisors, alumni, corporate recruiters, and students (from their co-op and internship experiences) as to development of industrial trends in computer science. In addition, all faculty regularly attend conferences to keep with the latest aspects of computer science. The department faculty meet annually for an all-day retreat at the beginning of the academic year, and weekly for an hour to an hour and a half during the year. At any of these meetings, any faculty member can propose changes to the computer science curriculum, which will then be discussed amongst all the faculty. Those changes that the department faculty approve are then brought to the institute's Curriculum Committee for consideration. Approval by the Curriculum Committee allows the department to implement the changes. New courses are usually introduced through the department's Special Topics in Computer Science course; those that prove popular are turned into regularly offered electives.

### D. Course Oversight

All courses in the department are taught by full-time faculty.

### E. Quality of Faculty Advising

Faculty who volunteer to be freshman advisers are required to attend a half-day workshop run by the Student Affairs Office on how to advise freshmen and what their special needs are. New faculty in the department are assigned another faculty member as a mentor. They meet regularly throughout the year to discuss all aspects of being a faculty member at Rose-Hulman and in Computer Science and Software Engineering, including student advising. New faculty members are not assigned students to advise until at least their second year at Rose-Hulman.

## 5. Curriculum

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> The program's requirements are consistent with its educational objectives and are designed in such a way that each of the POs can be achieved. The curriculum combines technical and professional requirements with general education requirements and electives to prepare students for a professional career and further study in the computing discipline associated with the program, and for functioning in modern society. The technical and professional requirements include at least one year of up-to-date coverage of basic and advanced topics in the computing discipline associated with the program. In addition, the program includes mathematics appropriate to the discipline beyond the pre-calculus level. For each course in the major required of all students, its content, expected performance criteria, and place in the overall program of study are published.
>
> Students have the following amounts of course work or equivalent educational experience.
>
> - Computer science: One and one-third years.
> - Math: One-half year that includes discrete mathematics. The additional mathematics might consist of courses in areas such as statistics, probability, calculus, linear algebra, numerical methods, number theory, geometry or symbolic logic.
> - Math and science combined: One year that includes a substantial laboratory science experience.
>
> (The Curriculum Criteria is specified in terms fractions of a year. Thirty semester hours generally constitutes one year of full-time study and is equivalent to 45 quarter-hours.)

### A. Title of Degree Program

| | |
|---|---|
| Transcript: | Computer Science |
| Diploma: | Computer Science |

### B. Credit Hour Definition

One quarter hour corresponds to one hour of lecture or three hours of laboratory per week. One academic year represents thirty weeks of classes, exclusive of final examinations.

### C. Prerequisite Flow Chart

The following figures illustrate the prerequisite structure of the program's named required courses, as well as the courses that may be used as computer science electives..

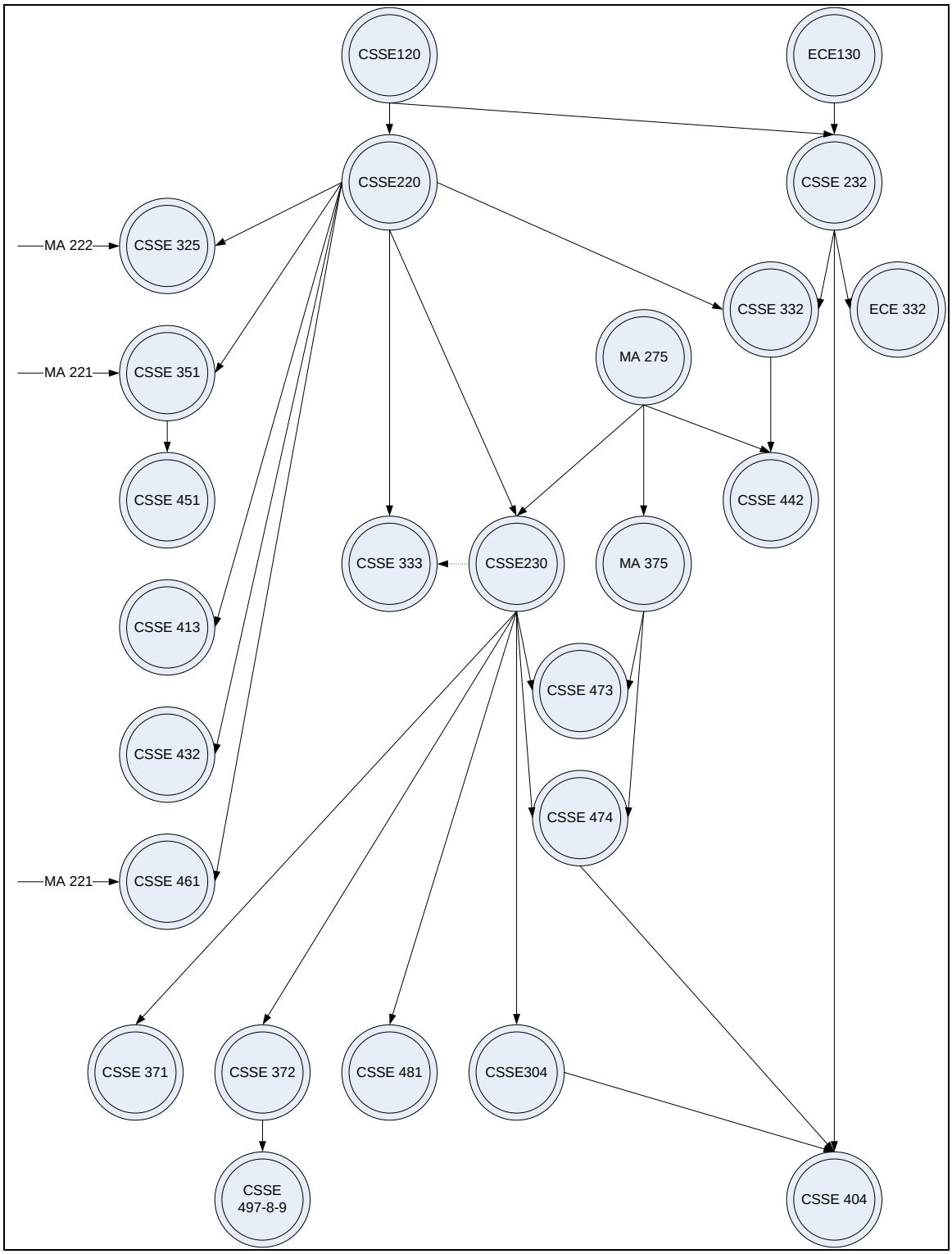



**Figure 3. Prerequisite relationships of selected CSSE and discrete mathematics courses.**

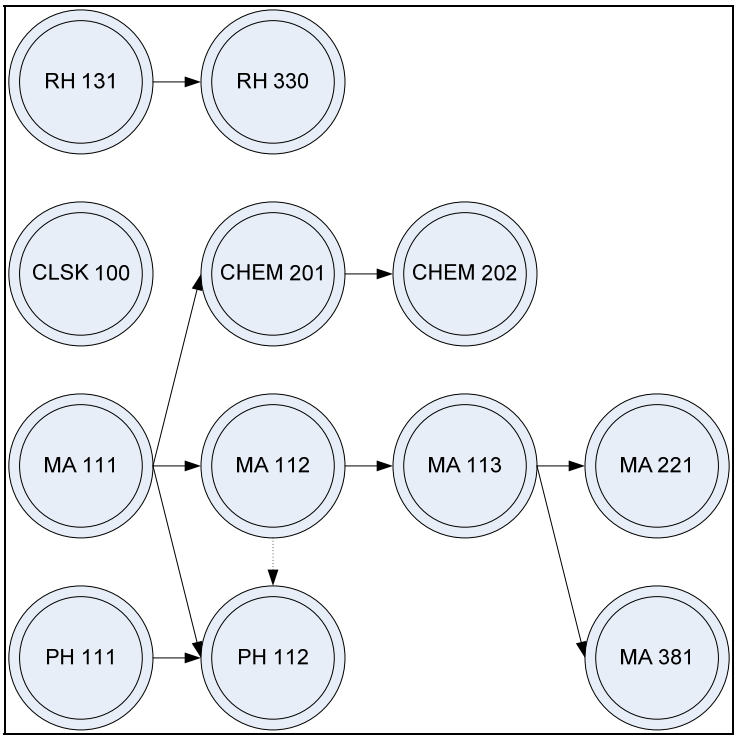



**Figure 4. Prerequisite relationships of other courses required for the computer science program.**

## D. Course Requirements of Curriculum (term by term and year by year)

**Table 11. Course Requirements (Freshman Year)**

| | | Category (quarter hours) | | | | | |
|---|---|---|---|---|---|---|---|
| Quarter | Course (Dept., Number, Title) | Program Core | Program Advanced. | Mathematics | Science | General Education | Other |
| First Quarter Freshman Year | CSSE 120 Fundamentals of Software Development I | 4 | | | | | |
| | MA 111 Calculus I | | | 5 | | | |
| | PH 111 Physics I | | | | 4 | | |
| | RH 131 Rhetoric & Composition | | | | | 4 | |
| | CLSK 100 College and Life Skills | | | | | | 1 |
| Second Quarter Freshman Year | CHEM 201 Engineering Chemistry I | | | | 4 | | |
| | MA 112 Calculus II | | | 5 | | | |
| | PH 112 Physics II | | | | 4 | | |
| | HSS Elective | | | | | 4 | |
| Third Quarter Freshman Year | ECE 130 Introduction to Logic Design | 4 | | | | | |
| | MA 113 Calculus III | | | 5 | | | |
| | HSS Elective | | | | | 4 | |
| | Science Elective | | | | 4 | | |
| SUBTOTALS | | 8 | 0 | 15 | 16 | 12 | 1 |

**Table 12. Course Requirements (Sophomore Year)**

| | | Category (quarter hours) | | | | | |
|---|---|---|---|---|---|---|---|
| Quarter | Course (Dept., Number, Title) | Program Core | Program Advanced. | Mathematics | Science | General Education | Other |
| First Quarter Sophomore Year | CSSE 220 Fundamentals of Software Development II | 4 | | | | | |
| | CSSE 232 Computer Architecture I | 4 | | | | | |
| | MA 221 Differential Equations and Matrix Algebra I | | | 4 | | | |
| | MA 275 Discrete & Combinatorial Algebra I | | | 4 | | | |
| Second Quarter Sophomore Year | CSSE 230 Fundamentals of Software Development III | 4 | | | | | |
| | CSSE 333 Database Systems | 4 | | | | | |
| | MA 375 Discrete & Combinatorial Algebra II | | | 4 | | | |
| | HSS Elective | | | | | 4 | |
| Third Quarter Sophomore Year | CSSE 304 Programming Language Concepts | 4 | | | | | |
| | ECE 332 Computer Architecture II | 4 | | | | | |
| | ECE/MA Elective | | | | | | 4 |
| | HSS Elective | | | | | 4 | |
| SUBTOTALS | | 24 | 0 | 12 | 0 | 8 | 4 |

**Table 13. Course Requirements (Junior Year)**

| | | Category (quarter hours) | | | | | |
|---|---|---|---|---|---|---|---|
| Quarter | Course (Dept., Number, Title) | Program Core | Program Advanced. | Mathematics | Science | General Education | Other |
| First Quarter Junior Year | CSSE 371 Software Requirements & Specification | 4 | | | | | |
| | CSSE Elective | | 4 | | | | |
| | MA 381 Introduction to Probability with Statistical Applications | | | 4 | | | |
| | HSS Elective | | | | | 4 | |
| Second Quarter Junior Year | CSSE 332 Operating Systems | 4 | | | | | |
| | CSSE 372 Software Project Management | 4 | | | | | |
| | ECE/MA Elective | | | | | | 4 |
| | RH 330 Technical Communications | | | | | 4 | |
| Third Quarter Junior Year | CSSE Elective | | 4 | | | | |
| | CSSE Elective | | 4 | | | | |
| | ECE/MA Elective | | | | | | 4 |
| | Free Elective | | | | | | 4 |
| SUBTOTALS | | 12 | 12 | 4 | 0 | 8 | 12 |

**Table 14. Course Requirements (Senior Year)**

| | | Category (quarter hours) | | | | | |
|---|---|---|---|---|---|---|---|
| Quarter | Course (Dept., Number, Title) | Program Core | Program Advanced. | Mathematics | Science | General Education | Other |
| First Quarter Senior Year | CSSE 497 Senior Project I or CSSE 494 Senior Thesis I | | 4 | | | | |
| | CSSE Elective | | 4 | | | | |
| | HSS Elective | | | | | 4 | |
| | Free Elective | | | | | | 4 |
| Second Quarter Senior Year | CSSE 498 Senior Project II or CSSE 495 Senior Thesis II | | 4 | | | | |
| | CSSE Elective | | 4 | | | | |
| | HSS Elective | | | | | 4 | |
| | Free Elective | | | | | | 4 |
| Third Quarter Senior Year | CSSE 499 Senior Project III or CSSE 496 Senior Thesis III | | 4 | | | | |
| | Free Elective | | | | | | 4 |
| | Free Elective | | | | | | 4 |
| | | | | | | | |
| SUBTOTALS | | 0 | 20 | 0 | 0 | 8 | 16 |
| TOTALS | | 44 | 32 | 31 | 16 | 36 | 33 |

The curriculum satisfies the CAC requirements for computer science programs as follows:

- 76 quarter hours of computer science (60 required)
- 31 quarter hours of mathematics, including discrete math (26 required)
- 47 quarter hours of mathematics and science (45 required)

## E. Course Descriptions

| **Computer Science and Software Engineering** | **CSSE 120** |
|---|---|
| | **Fundamentals of Software Development 1** |

### Catalog Description

This course introduces students to current practices of producing object-oriented software. Using a disciplined design process, students explore multi-threaded event-driven programming, the development of graphical user interfaces, and interaction among objects. Students design and implement software individually, in small groups, and in a challenging multi-week team project. This course presumes no prior programming experience.

### Instructor(s)

### Required/Elective

Required of majors in Computer Science, Software Engineering, Computer Engineering, Electrical Engineering, Physics, Mathematics and Economics

### Class/Laboratory Schedule

Three 105-minute sessions per week

### Prerequisites

None

### Textbook(s)

- Required textbook: *Interactive Programming in Java*, by Lynn Andrea Stein, 2004. Online and free to students.
- Supplementary textbook: *Big Java*, by Cay Horstmann, J. Wiley, 2006.

### Course Objectives

After successful completion of this course, students will:

1. Analyze and explain the behavior of simple programs involving:
    a. Fundamental non-object-oriented programming constructs, including:
        - Basic syntax and semantics of a higher-level language
        - Variables, types, expressions, and assignment
        - Simple I/O

- Conditional and iterative control structures
- Functions and parameter passing
- Arrays

b. Fundamental object-oriented programming constructs, including:
- Basic syntax and semantics of a class
- Fields, constructors and methods
- The distinction between primitive types/variables and non-primitive types/variables
- The relationship between the static structure of the class and the dynamic structure of the instances of the class (e.g. via Java's *new* operator)
- Visibility modifiers (e.g. Java's *public*, *protected* and *private*) and mutability modifiers (e.g. Java's *final*)
- The distinction between a static method/field and a non-static method/field
- Implementing an interface
- Extending a class
- Responding to an exception

2. Design, implement, debug and informally test simple programs in an object-oriented programming language, including programs that:
   a. Use the above fundamental programming constructs
   b. Support a graphical user interface developed by using a GUI toolkit
   c. Respond to user events
   d. Use threads
   e. Include "is-a" relationships among objects using a class hierarchy and inheritance
   f. Use large-scale API packages
3. Apply decomposition techniques, including:
   a. Using structured (functional) decomposition (breaking a function/program into subfunctions)
   b. Encapsulating behaviors and data into classes
   c. Using inheritance
4. Explain techniques that scale up to larger software development projects (and explain why they scale up), including:
   a. Pair programming
   b. Structured documentation (e.g., javadoc comments)
   c. Implementing by using stubs
   d. Implementing by using an iterative enhancement plan
   e. Using code conventions (e.g., Sun's code conventions)
   f. Decomposition techniques listed above
   g. Using an integrated development environment
   h. Using UML class diagrams
   i. Unit testing using a tool like JUnit
   j. Using simple metrics to evaluate quality of code
   k. Using simple team building and management techniques like brainstorming, roles and task lists
5. Work effectively as members of a small team

## Course Topics

Software development practices, fundamental object-oriented and non-object-oriented programming constructs. Also see *Schedule*, attached.

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

David Mutchler, CSSE 120 coordinator

| Computer Science and Software Engineering | CSSE 220<br>Fundamentals of Software Development II |
|---|---|

## Catalog Description

This course reinforces and extends students' understanding of current practices of producing object-oriented software. Students extend their use of a disciplined design process to include testing and space/time efficiency analysis. Students gain a deeper understanding of concepts from CSSE 120, including the use of inheritance, interfaces, polymorphism, abstract data types and encapsulation to enable software reuse and assist in software maintenance. This course introduces networking and database applications, and recursion as an example of functional programming. Students design and implement software individually, in small groups, and in a challenging multi-week team project.

## Instructor(s)

Wollowski, Mutchler, Bagert

## Required/Elective

Required.

## Class/Laboratory Schedule

Three 105 minute sessions each week, split into sections of lectures, demonstrations, student exercises, group and team work.

## Prerequisites

CSSE 120 – Fundamentals of Software Development I

## Textbook(s)

Weiss, Mark Allen: *Data Structures & Problem Solving using Java*, Third Edition
Gittleman, Art: *Advanced Java: Internet Applications*, Second Edition

## Course Objectives

After successful completion of this course, students will be able to:

1. Develop software that incorporates the following techniques:
    - Inheritance
    - Interfaces
    - Polymorphism
    - Casting
    - Function objects

    - Generics
    - Collections
2. Develop software in each of the following application areas:
    - Software that interacts with a database
    - Software that communicates over a network
    - Software that uses a graphical user interface
3. Perform the following steps of the software development cycle effectively:
    - Requirements elicitation
    - Design expressed as UML class diagrams
    - Implementation using an iterative enhancement plan and stubs/documentation before coding
    - Unit, system, and acceptance testing
4. Analyze the asymptotic worst, best, and average case run times of simple algorithms.
5. Select basic data structures (e.g, arrays, sequential lists, linked lists, stacks, queues, hash tables and trees) based on the time and space complexity of typical operations.
6. Develop software that uses recursive methods and data structures.

### Course Topics

- Inheritance
- Polymorphism
- Static overloading
- Dynamic binding
- Interfaces and abstract classes
- Method abstraction
- Functors
- Generic algorithms
- Iterators
- Recursion
- Analysis of algorithms
- Searching
- Sorting
- Datastructures, including: arrays, ArrayLists, Stacks, Queues, Linked Lists, Vectors, binary trees
- The following concepts of Software engineering: Requirements elicitation, use cases, acceptance testing
- Databases connectivity
- Networking

### Lab Topics

<see above>

### ABET Criteria

### Program Outcomes

**Professional Component**

**Prepared by**

Michael Wollowski, 04/01/06

**Computer Science and Software Engineering** **CSSE 221**

**Fundamentals of Software Development Honors**

## Catalog Description

This course is intended for students who have sufficient programming experience to warrant placement in an accelerated course covering the topics from CSSE 120 and CSSE 220. This course will satisfy the prerequisite requirements for courses that have CSSE 220 as a prerequisite. Students who receive a penalty grade in CSSE 221 may grade replace it with CSSE 220.

Students with a score of 4 or 5 on the Computer Science A or AB Advanced Placement exam may enroll in CSSE 221. Upon successful completion of CSSE 221 students will also be awarded 4 credits for CSSE 120. Furthermore, students with a score of 4 or 5 on the Computer Science AB Advanced Placement exam who complete CSSE 221 with a grade of C or better will be awarded a further 4 credits for CSSE 230.

## Instructor(s)

## Required/Elective

## Class/Laboratory Schedule

Three 105-minute sessions per week

## Prerequisites

This course is intended for students who have sufficient programming experience to warrant placement in an accelerated course covering the topics from CSSE 120 and CSSE 220. Students with a score of 4 or 5 on the Computer Science A or AB Advanced Placement exam are normally enrolled in CSSE 221. Other students may enroll with the instructor's permission.

## Textbook(s)

- Required textbook: *Interactive Programming in Java*, by Lynn Andrea Stein, 2004. Online and free to students.
- Required textbook: *Big Java*, by Cay Horstmann, J. Wiley, 2006.

## Course Objectives

After successful completion of this course, students will:

1. Meet all the learning outcomes of CSSE 120.
2. Meet all the learning outcomes of CSSE 220

## Course Topics

- **From CSSE 120:** Software development practices, fundamental object-oriented and non-object-oriented programming constructs. Developing software by working in teams.
- **From CSSE 220:** Inheritance; Polymorphism; Static overloading; Dynamic binding; Interfaces and abstract classes; Method abstraction; Functors; Generic algorithms; Iterators; Recursion; Analysis of algorithms; Searching; Sorting; Data structures, including: arrays, ArrayLists, Stacks, Queues, Linked Lists, Vectors, binary trees; The following concepts of software engineering: Requirements elicitation, use cases, acceptance testing; Databases connectivity; Networking.

## Lab Topics

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

David Mutchler, CSSE 221 coordinator

| Computer Science and Software Engineering | CSSE 230<br>**Fundamentals of Software Development III** |
|---|---|

## Catalog Description

This course reinforces and extends students' understanding of current practices of producing object-oriented software. Students extend their use of a disciplined design process to include formal analysis of space/time efficiency analysis (including recurrence relations and the big-Oh, Omega and Theta notations) and formal proofs of correctness (including mathematical induction). Students gain a deeper understanding of ideas from CSSE 220, including implementations of abstract data types by linear and non-linear data structures. This course introduces the use of randomized algorithms. Students design and implement software individually, in pairs, and in a team project.

## Instructor(s)

Boutell

## Required/Elective

Required.

## Class/Laboratory Schedule

Three 105 minute sessions each week, split into sections of lectures, demonstrations, student exercises, group and team work.

## Prerequisites

CSSE 220 or 221: Fundamentals of Computing II
MA 275: Discrete Mathematics I

## Textbook(s)

Weiss, Mark Allen: *Data Structures & Problem Solving using Java*, Third Edition

## Course Objectives

After successful completion of this course, students will be able to:

7. Describe the classical data structures (list, stack, queue, tree, priority queue, hash table, graph, set, dictionary) and understand the issues involved in implementation choices for each.
8. Describe the brute force, divide-and-conquer, greedy, dynamic programming, branch-and-bound and backtracking strategies, and recognize problems for which each strategy provides an appropriate solution.

9. Analyze and prove the asymptotic worst, best, and average case run times of algorithms, including non-trivial recursive algorithms, using Big O, little o, Omega, and Theta notation.
10. Analyze and prove lower-bound results on problems like comparison-based sorting and understand the implications of lower bound results.
11. Understand and analyze classical sorting, graph and tree-balancing algorithms.
12. Prove the correctness of simple algorithms, especially those involving loops and recursion.
13. Work effectively as part of a team.
14. Independently analyze problems, and design and implement solutions.

## Course Topics

- Binary Search Trees
- Tree iterators
- AVL trees
- Red Black trees
- AA trees
- Splay trees
- Hash tables
- Heaps
- Graph traversal algorithms
- Huffman codes
- Advanced sorting algorithms
- Sorting lower bounds
- Analysis of algorithms

## Lab Topics

<see above>

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Michael Wollowski, 04/01/06

| Computer Science and Software Engineering | CSSE 232<br>Computer Architecture I |
|---|---|

### Catalog Description

Computer instruction set architecture and implementation. Specific topics include historical perspectives, performance evaluation, computer organization, instruction formats, addressing modes, computer arithmetic, ALU design, floating-point representation, single-cycle and multi-cycle data paths, and processor control. Assembly language programming is used as a means of exploring instruction set architectures. The final project involves the complete design and implementation of a miniscule instruction set processor.

### Instructor(s)

### Required/Elective

Required.

### Class/Laboratory Schedule

Three 100-minute lecture and laboratory sessions each. The laboratory component of the class is incorporated into the regular lecture session. Students bring their laptop to class.

### Prerequisites

CSSE 120, ECE 130

### Textbook(s)

Patterson and Hennessy, "Computer Organization and Design," 3rd ed., Morgan Kaufman

### Course Objectives

After successful completion of this course, students will:

- Apply the principle of abstraction in analysis and design problems.
- Explain the binary representation of various forms of data.
- Given the syntax and semantics for an assembly language, design, implement, test, and debug simple programs in that language that involve arithmetic operations, input and output, various control structures (including selection, iteration, and recursive procedures), and interrupts.
- Design an instruction set architecture that is appropriate for a given application, taking into consideration key computer organization design principles.
- Design a hardware implementation of an instruction set architecture, and use modern computer aided design tools to model, simulate, test, and debug that

implementation.
- Analyze the performance of an instruction set architecture and implementation in terms of metrics such as CPU execution time, instruction count, cycles per instruction (CPI), clock frequency, throughput, and response time.
- Work effectively as members of a team.
- Communicate effectively in both written and verbal form.

**Course Topics**

- MIPS assembly language and machine language
- Data representation
- Computer arithmetic
- Register Transfer Language
- Datapath design
- Control unit design
- ALU design
- Evaluation of the performance of a computer architecture
- Use of SPIM and Xilinx software tools

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Archana Chidanandan, 7 October 2005

| Computer Science and Software Engineering | CSSE 304 Programming Language Concepts |
|---|---|

## Catalog Description

Syntax and semantics of programming languages. Grammars, parsing, data types, control flow, parameter passing, run-time storage management, binding times, functional programming and procedural abstraction, syntactic extensions, continuations, language design and evaluation. Students will explore several language features by writing an interpreter that implements them. individually, in pairs, and in a team project.

## Instructor(s)

Wollowski

## Required/Elective

Required.

## Class/Laboratory Schedule

Four 50 minute sessions each week.

## Prerequisites

CSSE 230 (Note that this course has MA215 as a prerequisite).

MA 315 recommended.

## Textbook(s)

*Essentials of Programming Languages Second Edition* by Friedman, Wand, and Haynes.

*The Scheme Programming Language, Third Edition* by R. Kent Dybvig

## Course Objectives

After successful completion of this course, students will be able to:

15. Design, implement, debug, and test complex programs in a functional programming language
16. Understand the role of:
    - Syntax specification
    - Syntax and semantics
    - Environments and continuations
    - Types
    - Abstraction
17. Understand the process of interpreting a program

18. Understand the relationship between syntax specification and implementation
19. Design, implement, debug, and test a complex interpreter

## Course Topics

- Backus-Naur Form
- Creating and evaluating functions in Scheme
- Lexical binding and scope
- Free and bound variables
- Abstract data types
- Parsing
- Environments
- Interpreters
- Lambda expressions
- Continuations
- Continuation passing style

## Lab Topics

None

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Michael Wollowski, 04/01/06

| Computer Science and Software Engineering | CSSE 325<br>Fractals and Chaotic Dynamical Systems |
|---|---|

## Catalog Description

Emphasis on the mathematical and computer graphics foundations behind fractal images and the relationship between chaotic dynamics and fractal geometry. Self-similar fractals, random fractals with Brownian motion, and fractals generated from dynamical systems. Fractal dimensions. Iterated Function Systems. Chaos in one-dimensional maps. Controlling chaos. Mandelbrot and Julia sets. Computer graphics. Same as MA 325.

## Instructor(s)

Laxer

## Required/Elective

Elective

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

CSSE 2220, Fundamentals of Software Development II

MA 222, Differential Equations and Matrix Algebra II

## Textbook(s)

*Chaos and Fractals: New Frontiers of Science* (Second Edition) by Peitgen, Jürgens, and Saupe. Springer-Verlag, 2004.

## Course Objectives

After successful completion of this course, students will be able to:

- Determine the dimension of a fractal object
- Determine the Hausdorff and Lebesgue measures of various sets
- Use iterated function systems and L-systems to model fractals
- Perform graphical analysis of dynamical systems
- Control chaos
- Determine mathematical properties of the Mandelbrot and Julia sets
- Generate fractal images via computer software

## Course Topics

Dimensions

Feedback and iterated function systems

The Chaos Game

L-systems

Brownian motion

Fractal landscapes

Chaos and horseshoe dynamics

Graphical analysis

Feigenbaum diagrams

Strange attractors

Julia sets

Mandelbrot set

## Lab Topics

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Cary Laxer, 3/28/06

| Computer Science and Software Engineering | CSSE 332<br>Operating Systems |
|---|---|

## Catalog Description

Students learn fundamental concepts of modern operating systems by studying how and why operating systems have evolved. Topics include CPU scheduling, process synchronization, memory management, file systems, I/O systems, privacy and security, and performance evaluation. Students implement parts of an operating system as a means of exploring the details of some of these topics.

## Instructor(s)

## Required/Elective

Required.

## Class/Laboratory Schedule

Three 100-minute lecture and laboratory sessions. The laboratory exercises are incorporated into the lecture sessions. Students bring their laptops to class. The class is held in D219 which is equipped with dual-bootable computers.

## Prerequisites

CSSE 220, CSSE 232

## Textbook(s)

W. Stallings, "Operating Systems: Internals and Design Principles", 5th ed., Prentice Hall

## Course Objectives

After successful completion of this course, students will:

- Explain the objectives and functions of modern operating systems.
- Describe how operating systems have evolved over time from primitive batch systems to sophisticated multiuser systems.
- Analyze the tradeoffs inherent in operating system design.
- Explain the hardware support necessary to implement protection mechanisms in operating systems.
- Analyze the performance of preemptive and non-preemptive scheduling algorithms.
- Explain how processes are created, managed and deleted by operating systems.

- Explain the need for process synchronization and demonstrate the proper use of common synchronization techniques.
- Explain memory hierarchy, including virtual memory, caching, paging and segmentation, and analyze memory performance and cost-performance tradeoffs.
- Explain the different approaches to file organization and their strengths and weaknesses.
- Design, implement, test, and debug software using a high-level language that uses explicit pointers, as well as explicit static and dynamic memory allocation.
- Work effectively in teams.
- Communicate effectively in writing.

**Course Topics**

- Introduction to UNIX and C programming
- History of operating systems
- Process model and thread models
- Concurrency issues such as synchronization, mutual exclusion, and deadlock.
- File systems
- Memory management
- Virtual memory
- Process scheduling
- Disk and I/O scheduling

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Archana Chidanandan, 7th October 2005

| Computer Science and Software Engineering | CSSE 333<br>Database Systems |
|---|---|

### Catalog Description

Relational database systems, with emphasis on entity relationship diagrams for data modeling. Properties and roles of transactions. SQL for data definition and data manipulation. Use of contemporary API's for access to the database. Enterprise examples provided from several application domains. The influence of design on the use of indexes, views, sequences, joins, and triggers. Physical level data structures: B+ trees and RAID. Survey of object databases.

### Instructor(s)

### Required/Elective

Required.

### Class/Laboratory Schedule

- Approx three 50 minute lecture / discussions per week
- Approx three 55-minute in-class lab and project sessions per week (the class is taught in a 3 by 1:45 studio format each week)
- Team presentation of projects (5 x 50 minutes each)
- Lab and project work also done outside class, average from 2004-5 class was approx 40 hours per student total.

### Prerequisites

CSSE 220 – Fundamentals of Software Development II, with a co-requisite of CSSE 230 – Fundamentals of Software Development III.

### Textbook(s)

*Fundamentals of Database Systems*, 4/E, by Ramez Elmasri and Shamkant B. Navathe. Addison-Wesley, 2004, ISBN: 0-321-12226-7. **Required.**

Self-Paced Training Kit: Microsoft SQL Server 2000 System Administration, Microsoft Press, ISBN: 0735619611, and Self-Paced Training Kit: Microsoft SQL Server 2000 Database Design and Implementation, Microsoft Press (this one's now out of print, replacement text is ISBN: 073561248X). **Both Suggested.**

### Course Objectives

After successful completion of this course, students will:

- An understanding of information models and systems
- A thorough understanding of Entity-Relationship model and Relational Model
- An awareness of other data modeling paradigms (such as legacy systems, object-oriented databases, etc.)
- An ability to design relational databases from problem statements
- An understanding of the basic aspects of physical database design
- An understanding and usage of a SQL-based query languages to access and modify data
- An ability to use database programming techniques such as stored procedures, triggers, etc.
- An understanding of information privacy, integrity, security, and preservation
- A basic understanding of Online Analytical Processing (OLAP) systems and Data Warehousing

## Course Topics

### Database Theory

- Modeling (ER, Relational, Normalization, Dependencies)
- Relational Algebra
- SQL, Transact-SQL, Embedded SQL
- Constraints
- Store Procedures
- Authorization
- Transactions
- Object Orientation
- XML
- OLAP/Data warehousing

### Database Practice

- Managing DB Files, Security
- Admin Tasks, Backing Up, Restoring
- Monitoring Performance
- Implementing DB Design
- Creating and Managing Databases, Data Types, Tables
- Data Integrity
- Indexes
- Views
- Stored Procedures
- Triggers
- Retrieving, Grouping
- Joining Tables
- Modifying Data

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Steve Chenoweth

| Computer Science and Software Engineering | CSSE 351<br>Computer Graphics |
|---|---|

### Catalog Description

Computer graphics algorithms, hardware and software. Line generators, affine transformations, line and polygon clipping, interactive techniques, perspective projection, solid modeling, hidden surface algorithms, lighting models, shading, and graphics standards. Programming assignments and a final project are required.

### Instructor(s)

Laxer, Mellor

### Required/Elective

Elective

### Class/Laboratory Schedule

Four 50-minute lectures per week.

### Prerequisites

CSSE 220, Fundamentals of Software Development II

MA 221, Differential Equations and Matrix Algebra I

### Textbook(s)

*Computer Graphics with OpenGL* (Third Edition) by Hearn and Baker. Pearson Prentice Hall, 2004.

### Course Objectives

After successful completion of this course, students will be able to:

- develop interactive computer graphics applications using a modern graphics API
- model geometric objects, curves, and surfaces
- project a three-dimensional scene onto a two-dimensional viewing surface
- apply lighting and shading to a scene
- describe the mathematical foundations of computer graphics (e.g. Bresenham's algorithm, transformation matrices, projection matrices)
- apply a texture map to an object

## Course Topics

Graphics systems

OpenGL

Graphics primitives and attributes

Geometric transformations

Interaction

Two- and three-dimensional viewing

Shading, lighting, and texture mapping

Curve and surface design

## Lab Topics

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Cary Laxer, 3/28/06

| Computer Science and Software Engineering | CSSE 371<br>Software Requirements and Specification |
|---|---|

### Catalog Description

Basic concepts and principles of software requirements engineering, its tools and techniques, and methods for modeling software systems. Topics include requirements elicitation, prototyping, functional and non-functional requirements, object-oriented techniques, and requirements tracking.

### Instructor(s)

### Required/Elective

Required.

### Class/Laboratory Schedule

Four 50-minute lectures each week

### Prerequisites

CSSE 230, Fundamentals of Software Development III

### Textbook(s)

- *Managing Software Requirements: A Use Case Approach*, Second Edition, by Dean Leffingwell and Don Widrig, ISBN 0-321-12247-X
- *Interaction Design: beyond human-computer interaction* by Jennifer Preece, Yvonne Rogers and Helen Sharp, ISBN 0-471-49278-7)

### Course Objectives

After successful completion of this course, students will:

Students who complete this course should be able to:

1. Understand the role of requirements engineering and its process
2. Use analysis techniques to develop a problem statement
3. Use multiple techniques to elicit requirements from stakeholders
4. Take the elicited requirements and develop a specification with functional and non-functional requirements

5. To be able to negotiate with the client and other stakeholders regarding priorities and scope
6. Use quality assurance techniques to verify that requirements are: verifiable, traceable, measurable, testable, accurate, unambiguous, consistent, and complete
7. Know how to manage requirements
8. To be able to develop and use user interface prototypes to validate requirements

**Course Topics**

- Problem Analysis
- Requirements Elicitation
- Development of Software Requirements
- Managing Scope of Requirements
- Quality Assurance of Requirements
- Requirements Management
- User Analysis
- User Interface Protoptyping

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Don Bagert; March 28, 2006

| Computer Science and Software Engineering | CSSE 372 Software Project Management |
|---|---|

## Catalog Description

Major issues and techniques of project management. Project evaluation and selection, scope management, team building, stakeholder management, risk assessment, scheduling, quality, rework, negotiation, and conflict management.

## Instructor(s)

## Required/Elective

Required.

## Class/Laboratory Schedule

Four 50-minute lectures each week

## Prerequisites

CSSE 230, Fundamentals of Software Development III

## Textbook(s)

*Rapid Development* by Steve McConnell, Microsoft Press, 1996, ISBN 1-55615-900-5.

## Course Objectives

After successful completion of this course, students will:

1. Explain several different lifecycle models for software engineering
2. Explain the fundamentals of software project management, including organizational structures and roles
3. Prepare a software project plan, including risk analysis
4. Lead or participate on a software project team, including sharing of responsibilities, building consensus, developing plans of action, and providing constructive feedback to peers
5. Plan, facilitate and participate in effective meetings
6. Participate as a member of a team that successfully monitors and controls a software project using traditional metrics
7. Conduct a software project retrospective
8. Explain the role of software engineering ethics and professional practices within the global society

9. Explain the role of motivation, satisfaction and lifelong learning in software professional careers
10. Use standard configuration management methods
11. Apply their understanding of configuration management in a software project team

**Course Topics**

- Software Methodologies
- Project Planning and Scheduling
- Risk Analysis
- Effective Project Meetings
- Project Team Coordination
- Project Tracking and Monitoring
- Software Metrics Selection and Usage
- Software Configuration Management
- Software Engineering Professionalism
- Software Engineering Ethics
- Software Engineering Professional Development

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Don Bagert; March 28, 2006

| Computer Science and Software Engineering | CSSE 404<br>Compiler Construction |
|---|---|

## Catalog Description

Theory and practice of programming language translation. Lexical analysis, syntax analysis, parser generators, abstract

syntax, symbol tables, semantic analysis, intermediate languages, code generation, code optimization, run-time storage

management, error handling. Students will construct a complete compiler for a small language.

## Instructor(s)

Clifton, Curt

## Required/Elective

Elective

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

CSSE 232, Computer Architecture I

CSSE 304, Programming Language Concepts

CSSE/MA 474, Theory of Computation

## Textbook(s)

*Engineering a Compiler* by Cooper and Torczon. Morgan Kaufmann, 2003.

## Course Objectives

After successful completion of this course, students will be able to:

- implement a program to tokenize an input stream based on a regular language definition
- convert a left-recursive context-free grammar into a right-recursive one
- construct the canonical set of LR(1) items and derive the tables used by an LR(1) parser for a given context-free grammar
- implement, using a parser-generator such as ANTLR and CUP, a program to parse a token stream and construct an intermediate representation based on a context-free

grammar
- implement code to perform context-sensitive analysis and type-checking on a program encoded in some intermediate representation
- implement code to construct symbol tables for a simple, imperative programming language
- implement a program that takes code written in a high-level language and generates corresponding code in a low-level target language, such as MIPS or Java Bytecode
- clearly explain selected, more advanced programming language implementation issues (such as optimization or adding non-main-stream features), based on independent study, and extend a working compiler to address them

## Course Topics

Regular Expressions

Nondeterministic Finite Automata

Deterministic Finite Automata

Scanner Generators

Context-free grammars

Backus-Naur Form

Top-Down Parsing (Eliminating Left-Recursion, Eliminating arbitrary look-ahead, First and Follow sets, Left-factoring)

Bottom-Up Parsing, Building LR(1) Tables

Parser Generators

Context-Sensitive Analysis (Attribute Grammars, Ad-hoc Syntax-Directed Translation)

Graphical Intermediate Representations (Abstract Syntax Trees, Directed Acyclic Graphs, Control Flow Graphs, Dependence Graphs)

Linear Intermediate Representations (Stack-machine code, Three-address Code, Static Single-Assignment Form)

Symbol tables

Type Systems and Practical Typechecking

Procedure abstraction

Code shape

Instruction scheduling (Tree-pattern matching, peephole optimization)

Redundancy elimination

## Lab Topics

## ABET Criteria

**Program Outcomes**

**Professional Component**

**Prepared by**

Cary Laxer and Curt Clifton, 6/6/06

| Computer Science and Software Engineering | CSSE 413<br>Artificial Intelligence |
|---|---|

## Catalog Description

Students investigate how to model and implement intelligent behavior using computers. Topics are chosen from how machines can: solve problems; reason and use knowledge; learn from experience; and perceive and act. Students explore these topics by implementing many of the ideas in software. Readings are drawn both from a textbook and from technical papers in recent conferences and journals.

## Instructor(s)

## Required/Elective

Elective

## Class/Laboratory Schedule

- Four 50 minute lecture / discussions per week
- During last 2 weeks, this becomes primarily lab time – Approx 7.5 hours.
- Lab work done primarily outside class, average from 2004 class was approx 60 hours per student total.

## Prerequisites

CSSE 220 – Fundamentals of Software Development II

## Textbook(s)

*Artificial Intelligence - A Modern Approach*, by Stuart Russell and Peter Norvig, **Second Edition.** Prentice Hall, 2003. **required**

## Course Objectives

After successful completion of this course, students will:

- Describe the fundamental design alternatives for AI and for intelligent agents.
- Use common tools for AI building systems.

- Develop a program to do heuristic search in a significant domain such as the TSP.[18][1]
- Build a knowledge representation model and use it for decision making.
- Create an AI program for which quality results and timeliness are both critical.
- Apply the concepts of learning and probability to AI.
- Understand the philosophical and ethical issues surrounding AI.

**Course Topics**

- Intelligent Agents
- Search Strategies, Heuristic Search & Local Search
- Constraint Satisfaction Problems
- Adversarial Search & Games
- Logical Agents
- First-order Logic, Inference, Chaining and Resolution
- Knowledge Representation, Reasoning Systems
- Planning Problems
- Acting Under Uncertainty and Learning from Observations
- Learning Decision Trees, Knowledge in Learning
- Inductive Logic
- Statistical Learning Methods
- Neural Networks
- Philosophical Foundations
- Ethics and Risks of AI
- AI: Present and Future, Discussion of Impact of AI

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Steve Chenoweth

| Computer Science and Software Engineering | CSSE 432<br>Computer Networks |
|---|---|

### Catalog Description

Organization, design, and implementation of computer networks, especially the Internet. Network protocols, protocol layering, flow control, congestion control, error control, packet organization, routing, gateways, connection establishment and maintenance, machine and domain naming, security. Each of the top four layers of the Internet protocol stack: application (FTP, HTTP, SMTP), transport (TCP, UDP), network (IP), link (Ethernet).

### Instructor(s)

Ardis

### Required/Elective

Elective.

### Class/Laboratory Schedule

Three 50-minute lectures and one 50-minute laboratory per week.

### Prerequisites

CSSE 220 - Fundamentals of Software Development II.

### Textbook(s)

*Computer Networking: A Top-Down Approach Featuring the Internet* (third edition) by James F. Kurose and Keith W. Ross, Addison Wesley, 2002, ISBN 0-321-22735-2.

### Course Objectives

After successful completion of this course, students will:

- Understand vocabulary of computer networking
- Understand networking services and how they are provided by multiple layers of protocols
- Understand principles and mechanisms of congestion control in networks.
- Understand principles and mechanisms of routing in networks.
- Understand security mechanisms for networks and why they are needed.
- Be able to implement reliable file transfer services on an unreliable network.

## Course Topics

Application layer of protocol stack: FTP, email, HTTP

Transport layer of protocol stack: UDP, TCP

Network layer of protocol stack: routing

Link layer of protocol stack: ethernet

Wireless networks

Network security

Network administration

## Lab Topics

Protocol frame formats

Sliding window protocols

Error detection

File transfer protocol

## ABET Criteria

This course primarily addresses ABET Criteria a, f, and k.

## Program Outcomes

This course primarily address Software Engineering Program Outcomes 4 and 5.

## Professional Component

Engineering Topic

This course provides 3 credits of engineering science, 0 credits of basic science, 0 credits of basic math, and 1 credit of engineering design.

A brief description (a couple of sentences) of what is done in this course that makes the above credit distribution seem reasonable.

## Prepared by

Mark A. Ardis, 3/23/06

# CSSE 442
# Computer Security

## Catalog Description

This course introduces ethical, theoretical, and practical issues of information security in computing systems. Implications of relevant professional codes of ethics are a recurring theme of the course. Foundational topics include access control matrices and standard system models, as well as policies for security, confidentiality, and integrity. Implementation issues include key management, cipher techniques, authentication, principles of secure design, representation of identity, access control mechanisms, information flow, life cycle issues, and formal evaluation and certification techniques. Additional topics include malicious logic, vulnerability analysis, and auditing. Computer network attack techniques are discussed and explored in a closed environment to motivate and inform discussion and exploration of computer network defense techniques.

## Instructor(s)

## Required/Elective

Elective.

## Class/Laboratory Schedule

Four 50-minute meetings per week, split roughly equally between lecture and in-class exercises.

## Prerequisites

CSSE 332 and MA 275

## Textbook(s)

*Computer Security: Art and Science*, Matt Bishop, Addison-Wesley, 2003.

## Course Objectives

Students who successfully complete this course should be able to:

- Apply ethical theory to cyberethics issues.
- Apply the Access Control Matrix model to represent protection systems.
- Explain the principles that guide the design of authentication mechanisms.
- Apply rudimentary mathematical, statistical, and protocol-based cryptosystem attacks.

- Analyze and apply basic techniques and protocols for cryptographic key management, cryptographic key exchange, public key interchange, and digital signature
- Analyze and apply cryptographic protocols in practical environments;
- Explain the eight basic design principles for security mechanisms;
- Apply basic access control mechanisms
- Analyze information flows in a system to determine whether or not they conform to policy.
- Discuss current events, practices, and tools relevant to computer security.

## Course Topics

- Cyberethics
- Security Law
- Access Control Matrices
- Authentication
- Security Policies
- Applications of Cryptography in Security
- Design Principles
- Access Control Mechanisms
- Information Flow
- Practical Aspects of Computer Network Attack and Defense

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Larry Merkle, October 9, 2005

| Computer Science and Software Engineering | CSSE 451<br>Advanced Computer Graphics |
|---|---|

### Catalog Description

Advanced topics in computer graphics. Topics will be drawn from current graphics research and will vary, but generally will include ray tracing, radiosity, physically-based modeling, animation, and stereoscopic viewing. Programming assignments and a research project are required.

### Instructor(s)

Laxer, Mellor

### Required/Elective

Elective

### Class/Laboratory Schedule

Four 50-minute lectures per week.

### Prerequisites

CSSE 351, Computer Graphics

### Textbook(s)

*Computer Graphics with OpenGL* (Third Edition) by Hearn and Baker. Pearson Prentice Hall, 2004.

### Course Objectives

After successful completion of this course, students will be able to:

- Implement a ray tracer.
- Read, understand, and discuss current professional journal articles in computer graphics.
- Research and learn a graphics API (other than OpenGL) and develop software that generates realistic images using that graphics API.

### Course Topics

Ray tracing

NURBS

Radiosity

Graphics APIs

**Lab Topics**

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Cary Laxer, 3/28/06

| Computer Science and Software Engineering | CSSE461<br>Computer Vision |
|---|---|

### Catalog Description

An introduction to 3D computer vision techniques. Both theory and practical applications will be covered. Major topics include image features, camera calibration, stereopsis, motion, shape from x, and recognition.

### Instructor(s)

Mellor

### Required/Elective

Elective.

### Class/Laboratory Schedule

Four 50-minute lectures per week.

### Prerequisites

CSSE 220 – Fundamentals of Software Development II

MA 221 – Differential Equations and Matrix Algebra I

### Textbook(s)

*Multiple View Geometry in Computer Vision*, Richard Hartley and Andrew Zisserman, 2003.

### Course Objectives

After successful completion of this course, students will:

1. Understand the properties of projective lines, projective planes, and projective spaces.

2. Be able to use homographies to transform image data (e.g. remove perspective distortion).

3. Understand the pin-hole camera model and projection matrices.

4. Understand epipolar geometry.

5. Be able to calibrate a camera.

6. Be able to reconstruct a 3D scene from image data.

## Course Topics

Image features

Projective geometry

Camera calibration

Stereopsis

Motion

Shape from *x*

## Lab Topics

none

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Matthew Boutell, Course coordinator, 3/29/2006

# CSSE 473

# Design and Analysis of Algorithms

## Catalog Description

Students study techniques for designing algorithms and for analyzing the time and space efficiency of algorithms. The algorithm design techniques include divide-and-conquer, greedy algorithms, dynamic programming, randomized algorithms and parallel algorithms. The algorithm analysis includes computational models, best/average/worst case analysis, and computational complexity (including lower bounds and NP-completeness). Same as MA 473.

## Instructor(s)

## Required/Elective

Elective.

## Class/Laboratory Schedule

Three 50-minute lectures and one 150-minute lab per week.

## Prerequisites

CSSE 230 and MA 375

## Textbook(s)

*Algorithms*, by R. Johnsonbaugh and M. Schaefer

## Course Objectives

After successful completion of this course, students will:

- Prove the correctness of selected algorithms.
- Use asymptotic and amortized analysis to prove upper, lower, and tight bounds on worst-case, average-case, and best-case time and space complexities of iterative and recursive algorithms.
- Use lower bounds and algorithmic analysis to prove the optimality of selected algorithms.
- Design algorithms to solve problems for which they are appropriate, using a variety of strategies (e.g. greedy, divide-and-conquer, backtracking, branch-and-

bound, probabilistic, and heuristic).
- Explain a variety of "advanced" algorithms (i.e. algorithms that have better asymptotic behavior than their naïve counterparts)
- Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data.
- Solve problems using fundamental graph algorithms, such as depth-first and breadth-first search, single-source and all-pairs shortest paths, transitive closure, topological sort, and at least one minimum spanning tree algorithm.
- Explain why some problems are undecidable, and discuss the implications.
- Define standard NP-Complete problems, including the Traveling Salesperson Problem, the Boolean Satisfiability Problem, the Bin Packing Problem, and the Subset-Sum Problem.
- Explain approximation algorithms for selected NP-Complete problems.
- Use reduction techniques to prove that selected problems belong to the complexity classes P, NP, NP-Hard, and NP-Complete.
- Explain the significance of the question "P=NP?"

**Course Topics**

- Fundamentals of Algorithmics, Analysis of Algorithms, and review
- Algorithm Design Techniques
  - Searching and its applications
  - Divide-and-Conquer
  - Sorting and Selection
  - Greedy Algorithms
  - Dynamic Programming
  - Text Searching
- Introduction to Computational Complexity

**ABET Criteria**

**Program Outcomes**

**Professional Component**

**Prepared by**

Larry Merkle, October 9, 2005

# CSSE 474
# Theory of Computation

## Catalog Description

Students study mathematical models by which to answer three questions: What is a computer? What limits exist on what problems computers can solve? What does it mean for a problem to be hard? Topics include models of computation (including Turing machines), undecidability (including the Halting Problem) and computational complexity (including NP-completeness). Same as MA 474.

## Instructor(s)

## Required/Elective

Elective.

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

CSSE 230 and MA 375

## Textbook(s)

*Introduction to Languages and the Theory of Computation*, 3rd Edition, John Martin, McGraw-Hill, 2003.

## Course Objectives

After successful completion of this course, students will:

- Discuss the concept of finite state machines.
- Design a deterministic finite-state machine to accept a specified language.
- Explain context-free grammars.
- Explain at least one algorithm for both top-down and bottom-up parsing.
- Explain the Church-Turing thesis and its significance.
- Determine a language's location in the Chomsky hierarchy (regular sets, context-free, context-sensitive, and recursively enumerable languages).
- Prove that a language is in a specified class and that it is not in the next lower class.
- Convert among equivalently powerful notations for a language, including

among DFAs, NFAs, and regular expressions, and between PDAs and CFGs.
- Explain how some problems have no algorithmic solution.
- Provide examples that illustrate the concept of uncomputability.
- Define the classes P and NP.
- Explain the significance of NP-completeness.
- Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it.

## Course Topics

- Regular Expressions and Finite Automata
- Nondeterminism and Kleene's Theorem
- Regular and Nonregular Languages
- Context-Free Grammars
- Pushdown Automata
- Context Free and Non-context-free languages
- Turing Machines
- Recursively Enumerable Languages
- Unsolvable Problems
- Measuring and Classifying Complexity
- Tractable and Intractable Problems

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Larry Merkle, October 9, 2005

| Computer Science and Software Engineering | CSSE 481<br>Web-based Information Systems |
|---|---|

## Catalog Description

In this course, students learn about several aspects of research: thinking creatively about interesting research problems, researching existing work in a chosen area, and keeping current in a field. Students are exposed to the process of research by writing a pre-proposal for a project that advances the web. Projects either develop new web-technologies or applications or investigate a topic of importance. Based on feedback received, groups of students write a research proposal which goes through a formal peer review process. Approved projects are pursued for the remainder of the quarter. Students present current research as well as give a final presentation of their group project. Selected web-technologies are introduced; in the past, these have included CGI programming and XML technologies.

## Instructor(s)

Wollowski

## Required/Elective

Elective.

## Class/Laboratory Schedule

Four 50 minute sessions each week.

## Prerequisites

CSSE 230

## Textbook(s)

None

## Course Objectives

After successful completion of this course, students will be able to:

20. Stay current in the area of the world-wide web.
21. Identify and propose applications which advance the web.
22. Develop applications and perform research which advances the web.
23. Present their own work in a scientific manner.

## Course Topics

- Identifying worthwhile projects
- Background research on current work
- Researching a project which advances the web
- Select current web-technologies
- Current and future developments of the web

## Lab Topics

None

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Michael Wollowski, 06/08/06

| Computer Science and Software Engineering | CSSE 497-8-9 Senior Project |
|---|---|

## Catalog Description

Group software engineering project requiring completion of a software system for an approved client. Tasks include project planning, risk analysis, use of standards, prototyping, configuration management, quality assurance, project reviews and reports, team management and organization, copyright, liability, and handling project failure.

**Note that this is a three-term, three-course senior project sequence.** It is an integral part of the Computer Science and Software Engineering majors at Rose-Hulman Institute of Technology. By working with companies on real problems, students learn key aspects of software development, including risk analysis and quality assurance.

## Instructor(s)

Ardis, Chenoweth (2005-6)

## Required/Elective

Required for SE major. Either this or Senior Thesis is required for CS major.

## Class/Laboratory Schedule

One 50-minute meeting with instructor each week. Additional team meetings and work by individuals and subgroups as required each week for the project. Twice per term, presentations are done by each team. Each team also meets with their client, potential users (if any) and other stakeholders as needed.

## Prerequisites

CSSE 371 (Requirements Engineering) and 372 (Project Management).

## Textbook(s)

None.

## Course Objectives

This course exercises many skills acquired in earlier courses. Students who successfully complete this course will be able to demonstrate their ability to:

**Communication Skills**

- **Reading:** read technical documents and offer constructive criticism of their

content and style
- **Writing:** write several different types of technical documents
- **Oral presentation:** prepare and deliver technical material at the appropriate level of detail

**Management Skills**
- **Leadership:** lead a small software team (if team leader) or ability to support the leadership of the team (if not)
- **Time management:** estimate and monitor personal time across multiple tasks
- **Meeting facilitation:** lead and participate in small groups in constructive meetings
- **Estimating:** estimate effort required to complete technical tasks
- **Risk:** assess project risks and plan mitigation strategies
- **Planning:** prepare a feasible plan for the accomplishment of several technical tasks
- **Monitoring:** track the progress of several tasks according to a plan

**Technical Skills**
- **Analysis:** analyze technical requirements and proposals for feasibility and to model the consequences of proposed solutions
- **Design:** construct appropriate abstractions of problems and solutions
- **Coding:** produce and inspect implementations of software according to project standards
- **Testing:** prepare test plans and to participate in both unit-level and system-level testing activities

**Professionalism**
- **Ethics:** identify and prevent unethical professional behavior
- **Intellectual property issues:** make appropriate professional judgments regarding choice of methods for protecting intellectual property
- **Social issues:** evaluate and avoid possible negative social aspects of a software product
- **Relationships with clients:** interact with clients in a professional manner

## Course Topics

In general the course topics depend upon the actual project being pursued by a team. The exception is that certain deliverables are expected and so there is learning or reinforcement of prior learning on how to do these, as needed. The ultimate criterion for some of those is the preferences client, they generally include making formal project presentations and delivery of some or all of the following project artifacts:

* Project Plan

* Configuration Management Plan

* Individual Weekly Reports

* Team Weekly Reports
* Meeting Agendas
* Meeting Minutes
* Requirements Specification
* Design Document
* Test Plan
* Test Case Specifications
* Test Results
* Software Manual

### Lab Topics

The majority of the above course topics are done in a lab or meeting room environment, as appropriate.

### ABET Criteria

This course primarily addresses ABET Criteria a, b, c, d, e, f, g, and h ?.

### Program Outcomes

This course addresses all the CSSE Program Outcomes.

### Professional Component

This course provides 0 credits of engineering science, 0 credits of basic science, 0 credits of basic math, and 12 credits of engineering design (4 credits per term). This is because the core of the whole sequence is applied analysis and design of engineering principles, as detailed in the technical objectives, *above*.

### Prepared by

Steve Chenoweth, 6/5/2006

# College and Life Skills CLSK 100

## Catalog Description

This course will assist the student in acquiring life skills & in learning more about themselves. These new skills will assist the student in a smooth transition from high school to college and will provide the students with the tools necessary for success as a student and in life. Additionally, this course will introduce students to people & resources at Rose-Hulman Institute of Technology who can assist them in providing a positive educational and personal experience.

## Instructor(s)

Tom Miller, curriculum coordinator (9 instructors each academic year)

## Required/Elective

Required.

## Class/Laboratory Schedule

One 50-minute lecture per week fall quarter, 10 class meetings

## Prerequisites

None

## Textbook(s)

None

## Course Objectives

- Assist a first-year student in a smooth transition from high school to college
- Course will introduce students to professional staff and resources on campus

## Course Topics

- Introduction to campus life/Greek life
- Office of the Registrar
- Electronic portfolio (Rose E-portfolio)
- Introduction to the Learning Center/resources

• Introduction to the Logan Library and research resources

• Introduction to Career Services

• Workshop on resume writing and eRecruiting

• Professional practices and experiences in professional engineering (Rose-Hulman Ventures)

• Introduction to Alumni Affairs and Student Alumni Association (SAA)

• Overview and results of the Myers-Briggs Personality Inventory

## Lab Topics

None

## ABET Criteria

## Program Outcomes

## Professional Component

This course provides 1 academic hour required for graduation.

## Prepared by

Tom Miller, 5/2/06

# Chemistry — CHEM 201

**Engineering Chemistry I**

## Catalog Description

Topics include stoichiometry, nomenclature, reactions in aqueous solution, atomic structure and periodic properties. Chemical bonding including Lewis dot structures and molecular geometry are considered in relation to the properties of solids, liquids, solutions and gases. Chemical kinetics is covered.

## Instructor(s)

DeVasher

## Required/Elective

Required.

## Class/Laboratory Schedule

Three 50-minute lectures and one 150-minute lab per week.

## Prerequisites

MA 111 – Calculus I

## Textbook(s)

*Chemistry*, 8th Edition, Chang, Raymond, McGraw-Hill, 2005.

## Course Objectives

After successful completion of this course, students will:

- be able to name simple ionic compounds and simple inorganic molecules correctly using systematic nomenclature
- possess an understanding of reaction stoichiometry, and predict the quantity of products being formed in a chemical reaction given a balanced chemical equation and known amounts of starting materials
- be able to take simple, accurate scientific measurements of matter (density, mass, volume, electrical conductivity) and propagate uncertainty in those measurements
- predict products of metathesis and combustion reactions
- be able to illustrate the structure of simple inorganic molecules using Lewis Dot Theory and predict the geometry of simple inorganic molecules using Valence Shell Electron

Pair Repulsion Theory

- posses and understanding of the relative rates of chemical reactions given experimental data

## Course Topics

Chemical nomenclature of simple ionic compounds and simple inorganic molecules

Stoichiometry

Gas Laws and Phase Changes

Dipole moments and Intermolecular forces

Valence Shell Electron Pair Repulsion Theory

Aqueous Phase Chemistry (metathesis reactions, oxidation-reduction reactions, combustion reactions)

Chemical kinetics

## Lab Topics

Weight Percentage and Density of a Solution

Thermal Decomposition of Sodium Hydrogen Carbonate

The Ideal Gas Law

Molecular Models

The Packing of Atoms

Solid State

Electrical Conductivity of Solutions

Qualitative Reduction-Oxidation (REDOX) Reactions

Determination of the Rate Constant & the Order of a Reaction

## ABET Criteria

## Program Outcomes

## Professional Component

This course provides 4 credits of engineering science, 0 credits of basic science, 0 credits of basic math, and 0 credit of engineering design.

This course is an introduction to chemistry as it applies to engineers.

**Prepared by**

Rebecca B. DeVasher, 03/17/06

# Chemistry CHEM 202

**Chemistry for Engineers II**

## Catalog Description

Equilibria for gas-phase, acid/base and redox reactions are covered, including electrochemistry. Thermochemistry and thermodynamics are discussed. Molecular orbital theory and nuclear chemistry are included

## Instructor(s)

DeVasher, Erwin, Jelski. McLean, Miller, Morris, Mottel, Mueller, Tilstra

## Required/Elective

Required.

## Class/Laboratory Schedule

Three 50-minute lectures and one 150-minute lab per week.

## Prerequisites

CHEM201 Engineering Chemistry I

## Textbook(s)

*Chemistry,* 8th Ed., R. Chang, McGraw-Hill, 2005

## Course Objectives

After successful completion of this course, students will:

- perform calorimetric measurements and computations
- record data in a tabular and graphical format with correct significant figures and uncertainties
- predict possible redox reactions
- know how to design a battery along with commercial applications
- be able to perform heterogeneous and homogeneous equilibria computations
- have an understanding of acid-base chemistry
- be able to perform conductometric, potentiometric, and indicator titrations

## Course Topics

- thermodynamics/thermochemistry
- electrochemistry (redox reactions)
- heterogeneous and homogeneous equilbria
- solubility product computations
- gas phase equilibrium
- acid-base chemistry

## Lab Topics

Qualitative Aspects of Chemical Energy
Calorimetry

Study of a Lead(II) Iodide System
Qualitative Aspects of Chemical Equilibrium
Visible Spectroscopy of Equilibrium
Conductometric, Potentiometric, and Indicator Titrations
Identification of an Unknown Acid, Base, or Salt Solution
Qualitative Analysis
Building Electrochemical Cells (Batteries)

## ABET Criteria

## Program Outcomes

None

## Professional Component

This course provides 4 credits of science.

This is a basic general chemistry science course.

## Prepared by

Michael R. Mueller, 3/30/06

# Electrical and Computer Engineering ECE130

**Introduction to Digital Logic**

## Catalog Description

Combinational logic design, Boolean algebra, logic minimization, Karnaugh maps, static and dynamic hazards, multiplexers and memories in combinational design, flip-flops, registers and counters, finite state machine design. Use of logic simulator for several design problems.

## Instructor(s):

Doering, Mu, Song

## Required/Elective:

Required

## Class/Laboratory Schedule

Four 50-minute lectures per week; 20% of class days devoted to mini-projects.

## Prerequisites:

None

## Textbook(s):

Marcovitz, A.B., *Introduction to Logic Design*, 2nd ed. McGraw Hill, 2005.

## Course Objectives:

After successfully completing this course the student should be able to:

- Appreciate the significance of digital information processing systems:
- Use binary representations of information:
- Analyze and design combinational circuits:
- Analyze and design sequential circuits:
- Analyze and design simple register-transfer level systems:
- Realize a circuit description as an FGPA-based system:

## Course Topics

- Binary numbers and arithmetic; 2's complement signed numbers
- Boolean algebra, logic gates, truth tables, Karnaugh maps
- Switch-level circuits, propagation delay, hazards and glitches
- MSI components (adder, subtractor, MUX, encoder, decoder, comparator)

- ROMs, PLAs, FPGAs
- Latches and flip-flops, timing diagrams
- Sequential circuit analysis, state tables, state diagrams
- Register, shift register, synchronous counter
- Finite state machine design, manual timing analysis
- Datapath design, controller/datapath design

## ABET Criteria

This course primarily addresses ABET Criteria a, b, e, g, k

## Program Outcomes

This course primarily addresses ECE Program Outcomes T1, T2, and H3

## Professional Component

This course provides 3 credits of engineering science, 0 credits of basic science, 0 credits of basic math, and 1 credit of engineering design.

## Prepared by

Ed Doering, 5/11/06

# Electrical and Computer Engineering ECE332

**Computer Architecture II**

## Catalog Description

ECE 332 Computer Architecture II 4R-0L-4C F,S Pre: CSSE 232: Pipelining, memory hierarchy, busses, instruction level parallelism, cost-performance tradeoffs, and review of new topics in areas of computer architecture or parallel processing. Team research project. Complements CS 332.

## Instructor(s)

Simoni, Yoder, Hudson

## Required/Elective

Required

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

CSSE 232

## Textbook(s)

D.A. Patterson and J.L. Hennessy. Computer Organization and Design: The Hardware/Software Interface, 3rd Ed. Morgan Kaufmann, 2005.

## Course Objectives

After successfully completing this course the student should be able to:

- Explain how pipelining can increase the performance of a processor.
- Explain the limitations of pipelining and how these limitations can be minimized. (for example: structural hazards and hardware fixes, data hazards and data forwarding, control hazards and branch prediction and predication, and exceptions and exception handling).
- Schedule a pipeline to minimize the effects of hazards. (for example, loop unrolling and software pipelining)
- Explain how memory affects system performance and the tradeoffs in designing a memory hierarchy (cache, etc.).
- Explain the difference between a superscalar and superpipelined processor and the advantages and disadvantages of each.
- Demonstrate an understanding of modern hardware/software techniques that are used to improve pipeline performance.

- Explain the tradeoffs inherent in multi-core processors and multiprocessor systems.
- Understand scholarly literature that is related to modern computer architecture and write a summary of it.

## Course Topics

- Pipelining
- Hazards: Data (RAW, WAW, WAR), Control, and Structural
- Loop unrolling and pipeline scheduling
- Superscalar pipelines and out-of-order execution
- Branch prediction
- SIMD instructions
- Software Pipelining
- IA-64 ISA: EPIC architecture, rotating registers, predication, speculative and advanced loads, register stacking
- Cache structure and memory hierarchy
- Multi-threaded and multi-core processing

## Lab Topics

None

## ABET Criteria

This course primarily addresses ABET Criteria a, c, g, i

## Program Outcomes

This course primarily addresses ECE Program Outcomes H1, H2, H3, T1, D&P

## Professional Component

This course provides 4 credits of engineering topics, 0 credits of basic science, 0 credits of basic math, and 0 credit of engineering design.

## Prepared by

Mario Simoni, 5/4/06

# Mathematics MA FTC

## Fast-Track Calculus

### Catalog Description

A 5-week fast paced course equivalent to Calculus I, II and III. Taught in the summer only to incoming freshmen. Review of differential calculus. Introduction to integration and the Fundamental Theorem of Calculus. Techniques of integration, numerical integration, applications of integration. L'Hopital's rule (and improper integrals). Separable first order differential equations, applications of separable first order differential equation. Series of constants, power series, Taylor polynomials, Taylor and McLaurin series. Vectors and parametric equations in three dimensions. Functions of several variables, partial derivatives, maxima and minima of functions of several variables, multiple integrals, and other coordinate systems. Applications of partial derivatives and multiple integrals.

### Instructor(s)

Various Mathematics instructors

### Required/Elective

Equivalent for MA111, MA112, MA113 for all majors

### Class/Laboratory Schedule

30 50-minute lecture/lab hours per week for five weeks

### Prerequisites

At least one year of high school Calculus, at least a 700 Math Score on the SAT (or equivalent), and approval by the Fast Track Selection Committee.

### Textbook(s) and other required material

Course notes

### Course Objectives

- Review with students differential calculus (including anti-derivatives) and vectors.
- Review with students the application of differential calculus and vectors in science and engineering.
- Introduce students to integral calculus (including elementary first order differential equations).
- Introduce students to the application of the integral calculus and differential equations in science and engineering.
- Introduce students to series of constants and functions, and the notions of

approximation and convergence.
- Introduce students to multivariable differential and integral calculus, and more vector techniques, especially in three dimensions
- Introduce students to the application of multivariable calculus and in science and engineering.
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

## Course Topics

**MA 111**

- Simple Maple commands, plotting, functions, modeling, implicit functions,
- parametric equations.
- Parametric equations
- inverse functions (trig, logs, exponentials)
- basic vectors (dot product, projections)
- Limits and continuity.
- rate of change, derivative
- tangent and normal lines.
- differentiation techniques review
- implicit differentiation
- Mean Value theorem, Extreme Value theorem
- concavity
- optimization problems
- Taylor Series
- Newton's Method
- Antidifferentiation,
- area and Riemann sums, the definite integral
- Related Rates

**MA 112**

- Integration techniques:
  - u-subsititutions
  - integration by parts.
  - partial fractions
- Numerical integration: trapezoidal rule, Simpson's Rule
- Applications:
  - volumes of revolution (disks)
  - volumes of revolution (shells)
  - arc length
  - surface area of revolution
  - work.
- Differential Equations:
  - separable differential equations
  - exponential growth
  - logistics equation.
  - Newton's Law of Cooling

    - falling body (variable acceleration)
    - Motion (Newton's Law), falling body with wind resistance, projectile
- Motion
    - mixing problems.
- Sequences and series:
    - telescoping series
    - geometric series
    - ratio test
    - integral test
    - power series and Taylor series
- Improper Integrals
- L'Hopital's rule
- First order linear ordinary differential equations,

**MA 113 topics**

- Vectors operations (cross product, triple scalar product)
- Equations of planes and lines in space
- Quadric surfaces
- Curvature, space curves
- Normal and Tangent vectors in three space
- Normal and Tangential components of acceleration
- Partial derivatives
- Implicit differentiation
- Derivatives of parametric equations, motion in the plane.
- Chain Rule
- Gradients
- Directional derivatives
- Taylor Series in two variables
- Tangent plane and normal lines
- Max/min in three space.
- Lagrange multipliers,
- Least squares.
- Double Integrals
- Applications of double integrals:
    - area (double integral)
    - Volume (double integral),
    - Center of mass (two-space)
    - centroid (two-space)
    - moments of inertia (two-space)
- Triple integrals, iterated triple integrals
- Applications of triple integrals
    - Center of mass (three-space)
    - moments of inertia (three-space)
- Polar coordinates and Area (polar coordinates)
- Cylindrical coordinates
- Volume in cylindrical coordinates (change of coordinate systems)
- Spherical Coordinates, volume in spherical coordinates.

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 15 credits of basic mathematics – equivalent to MA11, MA112, MA113

## Prepared by

S. Allen Broughton 5/1/06

# Mathematics

# MA 101
**Introductory Calculus**

# MA 102
**Differential Calculus**

**Note:** MA101/MA102 is a remedial sequence equivalent to MA111 for students who receive penalty (D or F) midterm or final grades for MA111

## Catalog Description

**MA101 Introductory Calculus:** Covers approximately the first half of MA 111, including analytic geometry in the plane, algebraic and transcendental functions, limits and continuity, and an introduction to differentiation. Entering first-year students will enroll in MA 111 and transfer to MA 101 if continuation of MA 111 is not appropriate.

**MA102 Differential Calculus:** Covers approximately the second half of MA 111, including the derivative, geometrical and physical applications of differentiation, and an introduction to integration and Fundamental Theorem of Calculus. Students who do not transfer to MA 101 in the fall quarter, but do not satisfactorily complete all of MA 111, may use their midterm grade in MA 111 for credit and grade in MA 101 and enter MA 102 at the beginning of the winter quarter.

## Instructor(s)

Various Mathematics instructors

## Required/Elective

Equivalent to the required course MA111

## Class/Laboratory Schedule

MA101 Five (5) - 50 minute sessions per week (5 weeks)
MA102 Five (5) - 50 minute sessions per week (10 weeks)

## Prerequisites

It is assumed that the student has a mastery of high school algebra, pre-calculus and trigonometry concepts.

## Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals*, 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

## Course Objectives (both courses combined)

- Introduce students to differential calculus and beginning integration, including anti-derivatives and the Fundamental Theorem of Calculus;
- Introduce students to the application of differential calculus and beginning integration in science and engineering
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

## Course Topics (both courses combined

- Functions and Pre-Calculus review
    - Graph of a function y=f(x), domain/range.
    - Properties of functions and graphs, e.g., increasing/decreasing intervals, local max/min.
    - Definition and properties of polynomial, trigonometric, exponential and logarithmic functions, and relevant inverse functions.
    - Parametric equations: physical interpretation as motion of a body

- Limits and Continuity
    - Limits.
    - Continuity
    - Intermediate value theorem
    -
- Differentiation
    - Average rate of change.
    - Instantaneous rate of change and definition of the derivative.
    - Formulas for elementary derivatives (polynomials, powers of $x$, sin($x$), cos($x$), tan($x$), sec($x$) , $e^x$, ln($x$), arcsin($x$), arctan($x$)
    - Linearity, product, quotient, and chain rules.
    - Implicit differentiation, parametric curves.
    - Approximation by secant and tangent lines, differentials
    - Mean value theorem, Rolle's Theorem , Extreme Value Theorem

- Applications of Derivatives
    - limits at infinity
    - Graphical/physical interpretation of first and second derivatives
    - Derivatives as velocity and acceleration, motion problems (including motion described by parametric equations.)
    - Optimization problems.
    - Related-rate problems.
    - Newton's method

- Integration
    - Position from velocity, area under a curve
    - Riemann sums
    - Fundamental Theorem, anti-derivatives and properties, specifically
    - linearity, polynomials, powers of $x$, $1/x$, sin($x$), cos($x$), $\sec^2(x)$, $e^x$, 1/sqrt($1-x^2$),

$1/(1+x^2)$.

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 5 credits of basic mathematics

MA101 – 2 credits

MA102 – 3 credits

## Prepared by

S. Allen Broughton 3/8/06

# Mathematics MA 111

**Calculus I**

## Catalog Description

Calculus and analytic geometry in the plane. Algebraic and transcendental functions. Limits and continuity. Differentiation, geometric and physical interpretations of the derivative, Newton's method. Introduction to integration and the Fundamental Theorem of Calculus.

## Instructor(s)

Various Mathematics instructors

## Required/Elective

Required.

## Class/Laboratory Schedule

Five (5) - 50 minute sessions per week

## Prerequisites

It is assumed that the student has a mastery of high school algebra, pre-calculus and trigonometry concepts.

## Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals* , 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

## Course Objectives

- Introduce students to differential calculus and beginning integration, including anti-derivatives and the Fundamental Theorem of Calculus;
- Introduce students to the application of differential calculus and beginning integration in science and engineering
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

## Course Topics

- Functions and Pre-Calculus review
    - Graph of a function y=f(x), domain/range.

    - Properties of functions and graphs, e.g., increasing/decreasing intervals, local max/min.
    - Definition and properties of polynomial, trigonometric, exponential and logarithmic functions, and relevant inverse functions.
    - Parametric equations: physical interpretation as motion of a body

- Limits and Continuity
    - Limits.
    - Continuity
    - Intermediate value theorem
    -
- Differentiation
    - Average rate of change.
    - Instantaneous rate of change and definition of the derivative.
    - Formulas for elementary derivatives (polynomials, powers of $x$, $\sin(x)$, $\cos(x)$, $\tan(x)$, $\sec(x)$ , $e^x$, $\ln(x)$, $\arcsin(x)$, $\arctan(x)$
    - Linearity, product, quotient, and chain rules.
    - Implicit differentiation, parametric curves.
    - Approximation by secant and tangent lines, differentials
    - Mean value theorem, Rolle's Theorem , Extreme Value Theorem

- Applications of Derivatives
    - limits at infinity
    - Graphical/physical interpretation of first and second derivatives
    - Derivatives as velocity and acceleration, motion problems (including motion described by parametric equations.)
    - Optimization problems.
    - Related-rate problems.
    - Newton's method

- Integration
    - Position from velocity, area under a curve
    - Riemann sums
    - Fundamental Theorem, anti-derivatives and properties, specifically
    - linearity, polynomials, powers of $x$, $1/x$, $\sin(x)$, $\cos(x)$, $\sec^2(x)$,
    - $e^x$, 1/sqrt(1-$x^2$), 1/(1+$x^2$).

### ABET Criteria

This course primarily addresses ABET criterion a.

### Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

### Professional Component

This course provides 5 credits of basic mathematics

**Prepared by**

S. Allen Broughton 3/8/06

# Mathematics MA 112

**Calculus II**

## Catalog Description

Techniques of integration, numerical integration, applications of integration. L'Hopital's rule and improper integrals. Separable first order differential equations, applications of separable first order differential equation. Series of constants, power series, Taylor polynomials, Taylor and McLaurin series.

## Instructor(s)

Various mathematics instructors

## Required/Elective

Required.

## Class/Laboratory Schedule

Five 50-minute lectures per week.

## Prerequisites

MA 111 - Calculus I or MA 102 - Differential Calculus or equivalent.

## Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals* , 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

## Course Objectives

- Introduce students to integral calculus (including elementary first order differential equations.
- Introduce students to the application of the integral calculus and differential equations in science and engineering
- Introduction students to series of constants and functions, and the notions of approximation and convergence
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems
- Develop student ability to communicate mathematically.

## Course Topics

- Integration---Basic Theory and Techniques
  - Riemann sums (review for Fall quarter freshman as needed)
  - Anti-derivatives for $x^n$ , $e^x$, $1/x$, $\sin(x)$, $\cos(x)$, $\sec^2(x)$, $1/(x^2+1)$, $1/\sqrt{1-x^2}$, $\cosh(x)$, and $\sinh(x)$
  - Linearity of integration
  - Integration by substitution
  - Integration by parts
  - Integration by partial fraction decomposition
  - Other integration techniques
  - L'Hopital's rule (may be taught in Calc I, review as needed)
  - Improper integrals
  - Numerical approximation using Trapezoidal and Simpson's rule

- Applications of Integration
  - Area
  - Displacement and distance travelled
  - Volumes of revolution (disk/shells)
  - Arc length, surface area of revolution
  - Work from force, potential energy

- Differential Equations
  - Definition, order, linearity
  - Separation of variables for separable first order equations.
  - Application to exponential growth and decay, population growth (logistic equation), Newton's law of cooling, salt tank problems, falling bodies (with and without air resistance.)
- Series
  - Series of constants
  - convergence
  - ratio test, integral test
  - power series
  - Taylor polynomials
  - Taylor and Mclaurin series of basic functions

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 5 credits of basic mathematics.

**Prepared by**

S. Allen Broughton 3/8/06

# Mathematics MA 113

**Calculus III**

## Catalog Description

Vectors and parametric equations in three dimensions. Functions of several variables, partial derivatives, maxima and minima of functions of several variables, multiple integrals, and other coordinate systems. Applications of partial derivatives and multiple integrals.

## Instructor(s)

Various Mathematics instructors

## Required/Elective

Required

## Class/Laboratory Schedule

Five (5) - 50 minute sessions per week

## Prerequisites

MA112 - Calculus II or equivalent.

## Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals*, 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

## Course Objectives

- Introduce students to multivariable differential and integral calculus, and more vector techniques, especially in three dimensions
- Introduce students to the application of multivariable calculus and in science and engineering.
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

## Course Topics

- Vectors in Three Dimensions, Vector-valued functions

    - Space coordinates and vectors in space
    - Dot product and projection (review from MA 111, maximum of one day)
    - Cross products
    - Lines and planes in space
    - Vector-valued functions
    - Differentiation/Integration of vector-valued functions
    - Curvature, arc length, unit tangent and normal vectors, components of acceleration

- Multivariable Differential Calculus
    - Functions of several variables
    - Partial derivatives
    - Chain rule
    - Directional derivatives, gradients
    - Tangent planes and normal lines
    - Unconstrained extrema
    - Lagrange multipliers

- Multivariable Integral Calculus
    - Double integrals, evaluation
    - Polar coordinates, change of coordinates
    - Triple integrals, evaluation
    - Cylindrical and spherical coordinates, integration in these coordinate systems

- Applications
    - Velocity and acceleration problems, projectile motion
    - Unconstrained and constrained multivariable max/min problems
    - Volume, surface area
    - Mass, moments, moment of inertia

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 5 credits of basic mathematics

## Prepared by

S. Allen Broughton 3/8/06

# Mathematics MA 221

**Differential Equations and Matrix Algebra 1**

## Catalog Description

Basic matrix algebra with emphasis on understanding systems of linear equations from algebraic and geometric viewpoints, including the least squares process and eigenvalues and eigenvectors. First order differential equations including basic solution techniques and numerical methods. Second order linear, constant coefficient differential equations, including both the homogeneous and non-homogeneous cases. Introduction to complex arithmetic, as needed. Applications to problems in science and engineering.

## Instructor(s)

Various mathematics instructors

## Required/Elective

Required

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

MA 113 Calculus III or equivalent or permission of mathematics department head.

## Textbook(s) and other required material

**Textbook***: Differential Equations & Linear Algebra with Boundary Values Problems* - Edwards/Rose-Hulman, 2006

**Computer Usage:** Maple

## Course Objectives

- Develop a deeper understanding of equations and their solutions, especially linear algebraic and differential equations.
- Improve mathematical modeling and analytical problem solving skills.
- Develop ability to communicate mathematically.
- Improve skill using the computer as a tool for mathematical analysis and problem solving.
- Introduce applications of mathematics, especially to science and engineering.

## Course Topics

- **First order differential equations**
  - Review basic notions (e.g., separation of variables, initial value problems)
  - Review $dx/dt = ax$ and $dx/dt = ax + b$ and show structure of solution as particular solution plus homogeneous solution
  - Numerical methods (e.g., Euler, RK4)
  - Applications as appropriate
- **Second order linear differential equations**
  - Constant coefficient, homogeneous case (solving the characteristic equation requires basics of complex arithmetic through Euler's formula)
  - Method of undetermined coefficients for non-homogeneous case
  - Method of variation of parameters for non-homogeneous case
  - Resonance
  - Applications as appropriate
  -
- **Matrix Algebra**
  - Matrix Arithmetic (e.g., addition, scalar multiplication, matrix multiplication, inverses --emphasis on the 2 x 2 case for "by-hand" inverting)
  - Understanding a matrix A as a transformation (avoiding general vector space ideas) ; $Ax = b$
  - Representation of systems of linear equations as matrix equations
  - Gaussian Elimination for solving $Ax = b$.
  - Structure of general solution for $Ax = b$: algebraic point of view (particular solution plus homogeneous solution);
  - Understanding the Least Squares Process: algebraic point of view (solve the normal equations); geometric point of view (project $b$ onto the range of $A$ and solve)
  - Eigenvalues and Eigenvectors: algebraic point of view (solve $Ax = \lambda x$ for $x$ and $\lambda$ ), and geometric point of view ($Ax$ is parallel to $x$)
  - Applications as appropriate.

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 4 credits of basic mathematics

**Prepared by**

S. Allen Broughton 3/8/06

# Mathematics MA 275

**Discrete and Combinatorial Algebra I**

## Catalog Description

An introduction to enumeration and discrete structures. Permutations, combinations and the pigeonhole principle. Elementary mathematical logic and proof techniques, including mathematical induction. Properties of the integers. Set theory. Introduction to functions.

## Instructor(s)

Various mathematics Instructors

## Required/Elective

Required for CS, SE, CPE

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

No Rose-Hulman course, but is assumed that the student has a mastery of high school algebra, pre-calculus and trigonometry concepts.

## Textbook(s) and other required material

Textbook: *Discrete and Combinatorial Mathematics, an Applied Introduction* 5th edition - Ralph P. Grimaldi

## Course Objectives

- Introduce student to discrete mathematics covering especially the topics noted below.
- Improve mathematical modeling and analytical problem solving skills, especially with discrete mathematics topics.
- Develop ability to communicate mathematically.
- Introduce applications of discrete mathematics, especially to science and engineering.

## Course Topics

- Logic
- Set Theory
- Proof Techniques
    - Induction

    - Proof by contradiction
    - Proof by contrapositive
- Counting Techniques
    - Permutations
    - Combinations
    - Pigeonhole principle
- Properties of integers
- Functions

### ABET Criteria

This course primarily addresses ABET criterion a.

### Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

### Professional Component

This course provides 4 credits of basic mathematics

### Prepared by

S. Allen Broughton 3/8/06

# Mathematics MA 375

**Discrete and Combinatorial Algebra II**

## Catalog Description

An introduction to enumeration and discrete structures. Permutations, combinations and the pigeonhole principle. Elementary mathematical logic and proof techniques, including mathematical induction. Properties of the integers. Set theory. Introduction to functions.

## Instructor(s)

Various mathematics Instructors

## Required/Elective

Required for CS, SE

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

MA275 – Discrete and Combinatorial Algebra I

## Textbook(s) and other required material

Textbook: *Discrete and Combinatorial Mathematics, an Applied Introduction* 5th edition - Ralph P. Grimaldi

## Course Objectives

- Continue to introduce students to discrete mathematics, covering especially the topics noted in the course topics below.
- Improve mathematical modeling and analytical problem solving skills, especially with generating functions and recurrence relations.
- Develop an ability to communicate mathematically.
- Introduce applications of discrete mathematics, especially in science and engineering.

## Course Topics

- Languages and Finite State Machines
    - Minimization of finite state machines
- Relations
    - Equivalence relations

    - Partial orders and Hasse diagrams
    - Relation matrices
    - Modular arithmetic

- Principle of Inclusion-Exclusion
- Generating functions ordinary and exponential
- Recurrence relations

### ABET Criteria

This course primarily addresses ABET criterion a.

### Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

### Professional Component

This course provides 4 credits of basic mathematics

### Prepared by

S. Allen Broughton 3/8/06

# Mathematics MA 381

**Introduction to Probability with Applications to Statistics**

## Catalog Description

Introduction to probability theory; axioms of probability, sample spaces, and probability laws (including conditional probabilities). Univariate random variables (discrete and continuous) and their expectations including these distributions: binomial, Poisson, geometric, uniform, exponential, and normal. Introduction to moment generating functions. Introduction to jointly distributed random variables. Univariate and joint transformations of random variables. The distribution of linear combinations of random variables and an introduction to the Central Limit Theorem. Applications of probability to statistics.

## Instructor(s)

Various Mathematics instructors

## Required/Elective

Required for CS, SE, EE, CPE, replaces MA223

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

MA 113 Calculus III

## Textbook(s) and other required material

*Fundamentals of Probability with Stochastic Processes*, 3rd Edition, by Saeed Ghahramani.

## Course Objectives

- An understanding of the ideas of probability and probability modeling, including sample spaces, axioms, discrete and continuous random variables, univariate and joint distributions, moment generating functions, the central limit theorem, and basic statistical inference
- An understanding of the special language, notation, and point of view of probability
- An understanding of the concepts of probability necessary to undertake basic modeling and decision making in math, science, and engineering
- The ability to solve standard computational problems in probability, which includes using the computer as a tool for mathematical analysis and problem solving
- An understanding of the relationship between random variables and their distribution

functions
- The ability to recognize special models, such as Bernoulli trials or Poisson processes
- An understanding of how probability is applied to inferential statistics
- The ability to communicate in mathematical terms
- Using simulation as a tool in approximating probabilities
- Preparing for the types of problems you may see if you take the first actuarial exam

## Course Topics

- Axioms of Probability: Sample space, events, axioms of probability, basic theorems
- Combinatorial Methods: Basic Counting Principles, Combinations, Permutations
- Conditional Probability and Independence**:** Conditional probability, laws of multiplication and total probability, Baye's formula, independence
- Distribution Functions and Discrete Random Variables: Random variables, distribution functions, discrete random variables, expectations of discrete random variables, variances and moments of discrete random variables, standardized random variables.
- Special Discrete Distributions**:** Bernoulli and binomial, Poisson (including Poisson process), and geometric distributions
- Continuous Random Variables: Probability density functions, density function of a function of a random variable, expectations and variances
- Special Continuous Distributions: Uniform, normal, exponential distributions
- Chapter 8. Bivariate Distributions**:** Joint distributions, independent random variables, conditional distributions
- More expectations and variances: Expected values of sums of random variables, covariance, correlation, conditioning on random variables
- Sums of independent random variables: Moment generating functions, sums of independent random variables, Markov and Chebyshev inequalities, central limit theorem

## ABET Criteria

This course primarily addresses ABET criterion a.

## Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH7 and RH8.

## Professional Component

This course provides 4 credits of basic mathematics

## Prepared by

S. Allen Broughton 3/8/06

| Physics and Optical Engineering | PH 111 Physics I |
|---|---|

### Catalog Description

Newton's laws of motion, gravitation, Coulomb's law, Lorentz force law, strong and weak nuclear forces, conservation of energy and momentum, torque and angular momentum, relevant laboratory experiments.

### Instructor(s)

Sudipa Kirtley

### Required/Elective

Required.

### Class/Laboratory Schedule

Four 50-minute lectures per week and one 160-minute lab every other week

### Prerequisites

None

### Textbook(s)

*Physics for Scientists and Engineers,* R. Knight, Pearson-Addison & Wesley, 2004.

### Course Objectives

After successful completion of this course students will:

Calculus-based introductory treatment of fundamental laws of mechanics with emphasis on development of problem-solving skills.

### Course Topics

A. Vectors: Scalars and vectors, decomposition of vectors into components, addition of vectors, cross and dot products, unit vectors.

B. Kinematics: Position, displacement, average and instantaneous velocity and acceleration equations for vertical and horizontal motions, definition of period, angular velocity and acceleration, relation between linear and angular terms, centripetal and tangential accelerations.

C. Newton's laws of motion with application. Forces and Newton's Laws: Forces between two surfaces, Newton's three laws, calculation of resultant acceleration, calculations involving light strings and pulleys, coefficients of static and dynamic friction.

D. Gravitation: Newton's law of gravity, universal gravitational constant, how to find the mass of the sun and the period of earth's orbit.

E. Electric Field and Coulomb's Law: Electric charges, electric potential, equipotential lines, lines of force, force on a charge in an electric field.

F. Magnetism: Magnetic field, lines of force and equipotential lines for a magnetic field, Earth has a magnet, force on a moving charge in a magnetic field.

G. Momentum, Impulse, and Collisions: Definitions, Newton's second law in momentum form, law of conservation of overall momentum, elastic and inelastic collisions, one- and two-dimensional collisions.

H. Work: Work done by a force from a force-distance graph, definition of kinetic energy, and its relation to the work done by an accelerating mass. Potential energy, conservation of mechanical energy in the absence and presence of dissipatory forces.

### Lab Topics

- Analysis of the motion of falling bodies, including acceleration due to gravity and terminal velocity. Sonic Ranger and PC computer used for data collection and analysis. (1 week)
- Conservation of momentum using an air-track. Sonic Ranger and PC computer used for data collection and analysis. (1 week)
- Newton's Second Law using falling mass connected to an air-track glider via a string draped over a pulley. Sonic Ranger and PC computer used in data collection and analysis. (1week)
- Analysis of pendulum period and maximum velocity as a function of amplitude of swing, shaft encoder and PC computer used in data collection and analysis. (1 week)

### ABET Criteria

This course primarily addresses ABET criteria a, b, and g.

### Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH6, RH7, RH8 and RH9.

### Professional Component

This course provides 0 credits of engineering science, 4 credits of basic science, and 0 credits of engineering design.

**Prepared by**

Sudipa Kirtley, 2005

| Physics and Optical Engineering | PH 112 Physics II |
|---|---|

### Catalog Description

Oscillations, one-dimensional waves, introduction to quantum mechanics, electric fields and potentials, electric current and resistance, DC circuits, capacitance, relevant laboratory experiments.

### Instructor(s)

Charles Joenathan

### Required/Elective

Required.

### Class/Laboratory Schedule

Four 50-minute lectures per week and one 160-minute lab every other week

### Prerequisites

PH111, Physics I and MA 111; Co-requisite : MA 112

### Textbook(s)

*Physics for Scientists and Engineers,* R. Knight, Pearson-Addison & Wesley, 2004.

### Course Objectives

After successful completion of this course students will:

Calculus-based introductory treatment of fundamental laws of mechanics with emphasis on development of problem-solving skills.

### Course Topics

A. Torque, moment of inertia, angular momentum as used in the analysis of rotational motion. Equations of motion in terms of angular variables, Linear and angular variables, Rotational kinetic energy and rotational inertia, Parallel-axis theorem, Torque, moment arm, line of action of a force, Newton's second law in angular form, Work and rotational kinetic energy, Rolling bodies, KE in terms of center of mass, Angular momentum of a system of particles, and of a rigid body, Conservation of angular momentum.

B. Oscillatory motion. Simple harmonic motion: frequency, period, amplitude, angular frequency, wave number, phase. Velocity and acceleration amplitudes, Linear oscillator, Energy, Pendulums

C. Vibrations and sound waves. Transverse and longitudinal waves, Equation of a traveling wave, Wave speed on a stretched string, Power, Superposition of waves, Interference of waves, Standing waves and resonance.

D. Coulomb's Law and its applications. Electric field and field lines, E due to a system of point charges and a continuous charge distribution, Force and field, Dipole, and dipole in the presence of E (torque, and PE)

E. Electric field Gauss's Law. Electric flux, Application of G's Law to find the E due to a line charge, a spherical charge and a cylindrical distribution of charge, E inside a conductor

F. Electric potential Electric potential energy, Work done by electric field, Finding V from E, Finding E from V, Finding V for a group of charges and a continuous distribution of charges

G. Capacitance and dielectrics Capacitors, and capacitance. Finding C for different capacitors, Capacitors in parallel and series, Potential energy stored in a capacitor, Capacitors and dielectrics

H. Current, resistance and electromotive force. Electric current, Current density, drift velocity, charge carriers, Resistance and resistivity, Ohm's Law, Power

I. Direct current circuits. Emf, Kirchhoff's Laws, Resistances in series and parallel, Multiple loop circuits, RC Circuits (charging and discharging a capacitor, behavior of V, q, and I as a function of time in the two cases)

J. Electromagnetic radiation. Origin of EM radiation and the broad form of the spectrum. Definition and examples of black body radiation, Wien's and Stefan's Laws, Planck's solution for blackbody radiation.

K. Electromagnetic waves as particles: Photoelectric Effect and Compton Scattering, Particles as Waves, interference, wavelength of a particle. Uncertainty Principle, brief description of tunneling, quantization of energy

## Lab Topics

- Investigation of torques and moment of inertia through measurement of angular acceleration of a rotating plate
- An investigation of the characteristic resonance frequencies of a string under tension (held at both ends), and a tube (open at one end).
- Study of current-voltage relationship in a DC circuit. Application of Kirchhoff's laws.

- Study the charging of a capacitor in an RC-circuit using ADC and PC-compatible computer.

### ABET Criteria

This course primarily addresses ABET criteria a, b, g.

### Program Outcomes

This course primarily addresses RHIT Institutional Outcomes RH6, RH7, RH8 and RH9.

### Professional Component

This course provides 0 credits of engineering science, 4 credits of basic science, and 0 credits of engineering design.

### Prepared by

Charles Joenathan, 2005

## Catalog Description

**RH 131 Rhetoric and Composition 4R-0L-4C F,W,S**
Examines selected pieces of writing which are used as models for student composition. Emphasizes the use of evidence and methods of argumentation. Required of all students. (May not be counted for an Area Minor in Language and Literature.)

## Instructor(s)

Carlson, Carvill, Dyer, Minster, Smith

## Required/Elective

Required.

## Class/Laboratory Schedule

Four 50-minutes periods per week.

## Prerequisites

None.

## Textbook(s)

Varies by instructor.

## Course Objectives

- to identify, evaluate, and construct effective arguments;
- to produce a clean, efficient, and grammatically correct style
- to adapt to a variety of rhetorical situations and audiences;
- to read critically;
- to conduct research and document sources

## Course Topics

Rhetorical Strategies: Audience Analysis, Genre, Voice, etc.

Process: Prewriting, Drafting, Revising, Editing

Research Methods, Documentation, and Assessment of Evidence

Critical Reading

## Lab Topics

None.

## ABET Criteria

## Program Outcomes

## Professional Component

## Prepared by

Caroline Carvill, March 2006

| Humanities and Social Sciences | RH 330 Technical Communication |
|---|---|

## Catalog Description

**RH 330 Technical Communication 4R-0L-4C F,W,S Pre: RH 131 and Junior class standing or consent of instructor**

Discusses the preparation and presentation of engineering reports, both oral and written. (May not be counted for an area minor in Language and Literature.)

## Instructor(s)

House, Minster, Watt, Williams, and Zoetewey

## Required/Elective

Required.

## Class/Laboratory Schedule

Four 50-minute lectures per week.

## Prerequisites

RH 131 – Rhetoric and Composition

## Textbook(s)

Laura J. Gurak and John M. Lannon, *A Concise Guide to Technical Communication*. 2nd Edition. New York: Pearson Longman, 2004

Edward R. Tufte, *The Cognitive Style of PowerPoint*. Cheshire, CT: Graphics Press LLC, 2003

## Course Objectives

After successful completion of this course, students will understand:

- writing in context, including analyzing professional cultures in technical fields and professions; analyzing social contexts and audiences to determine how they shape the various purposes and forms of writing; communicating the technical aspects of the student's field or discipline to non/specialists; and understanding the ethical implications of working within the nexus of technology and culture.
- writing process, including developing strategies for planning, researching, drafting, revising, and editing documents that respond effectively and ethically to technical situations and audiences and developing strategies for managing short and long term

projects.

- collaboration, including learning and applying strategies for effective collaboration within and across teams with overlapping interests; developing effective models for distribution of labor; communicating orally, in writing, and electronically with colleagues within and across hierarchies; negotiating conflict; and responding constructively to peers' work.
- research, including field research (e.g. analyzing the needs of an audience and supplying appropriate technical information to meet their needs) and library research (e.g. locating non/technical information and properly documenting it in the student's own work).
- document design, including designing usable, clear, persuasive, and accessible documents that meet multiple user and reader needs and learning to interpret and argue with visual data.

## Course Topics

Audience analysis; discourse community analysis; rhetoric and persuasion; ethics in communication; summary writing; memo writing; peer review strategies; team communication; intercultural communication; case study analysis; elements of page design/layout; effective use of visuals; fundamentals of effective writing (grammar, spelling, punctuation, mechanics).

## Lab Topics

Not applicable.

## ABET Criteria

This course primarily addresses ABET Criteria g.

## Program Outcomes

RH 6

## Professional Component

Not applicable.

## Prepared by

Julia M. Williams, 5/16/06

## 6. Technology Infrastructure

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> Computing resources are available, accessible, systematically maintained and upgraded, and otherwise adequately supported to enable students to achieve the program's outcomes and to support faculty teaching needs and scholarly activities. Students and faculty receive appropriate guidance regarding the computing resources and laboratories available to the program.

### A. Computing Resources

**Resources used for instruction**

***Institutional computing resources:***

Rose-Hulman's Information Technology system is a true, enterprise computing system. As indication of institutional commitment to educational excellence, the Board of Trustees in 2001 provided a $700,000 budget advance to completely replace our network.

Three large router/switches are the primary campus networking infrastructure components and are configured into a triangular interconnection to provide redundant paths. All major buildings are connected to these three devices via gigabit links (star topology). Approximately 250 edge-switches are located around campus and connected to the gigabit links. The edge-switches provide 100 Mbits/sec links to each individual port on campus. Table 15 provides a summary of the total number of active network ports on campus. Approximately 6,000 network ports serve a population of 2,000 users.

In 2003, IAIT added a third StorageTek L80 robotic tape library to the production computing environment. This tape library utilizes new tape drive technology to store more data (160 GB versus 40 GB) at high transfer rates. At the same time, one of the existing StorageTek L80 tape libraries was upgraded to include a fiber channel interface. The Rose-Hulman computing environment now includes a storage area network with a four TB disk array, two 6-drive, 80-cartridge tape libraries, and six servers. The fiber channel network enables direct data transfers between devices so that the disk array and tape libraries do not have to be directly attached to any of the servers.

IAIT also manages a complete mirror system known as the Myers Disaster Recovery Facility. The production database is automatically mirrored to the database server located in Myers Hall. The pre-production database is also located on the Myers database server. The Disaster Recovery Facility includes replicas of the production Kerberos, LDAP and Microsoft Active Directory data stores, thus, enabling a rapid transition to the backup systems. Also in 2003, the Institute upgraded the Olin 203 Public Laboratory with 33 HP xw4000 personal computers. These are well-configured systems that will provide an acceptable computing environment for at least three years.

In 2004, the Institute negotiated a new contract for our Internet connection increasing the bandwidth to 45 Mbs, three times the previous bandwidth. In 2005, the Institute purchased and installed a storage Tek LSA200 disk array. This device added two terabytes disk capacity, for a total of four terabytes, for the Microsoft Exchange cluster and other Microsoft Window servers.

Currently all academic buildings are covered by IEEE 802.11b wireless network (over 40 access points). This wireless deployment also incorporates VoIP (voice over internet protocol). In addition, Table 15 shows the placement of 6036 active fiber connections at the Institution.

**Table 15. Placement of active fiber connections.**

| Campus Connections | | |
|---|---|---|
| | | **Active Network Connections** |
| Crapo Hall | | 521 |
| Facilities | | 29 |
| Hadley Hall | | 39 |
| Hatfield Hall | | 143 |
| Hulman Union | | 160 |
| Logan Library | | 249 |
| Moench Hall | | 1,480 |
| Myers Hall | | 215 |
| Observatory | | 24 |
| Observatory Lodge | | 18 |
| New Olin Hall | | 520 |
| Old Olin Hall | | 520 |
| Public Safety | | 14 |
| Chi Omega House | | 10 |
| ROTZ Lab | | 29 |
| SRC | | 45 |
| White Chapel | | 6 |
| Residence Halls - *Totals* | *2,014* | |
| Blumberg Hall | | 83 |
| Mees Hall | | 83 |
| Scharpenberg Hall | | 80 |
| Deming Hall | | 143 |
| BSB Hall | | 184 |
| Speed Hall | | 168 |
| New Residence Hall | | 662 |
| Skinner Hall | | 71 |
| New Apartment Res. | | 540 |
| | | 6,036 |

This network is stable and reliable, allowing faculty to confidently include network-based instructional materials in their courses. In addition to outstanding web pages prepared by individual faculty members, their efforts are now supported by a campus-wide course management system.

All members of the campus community have access to a wealth of locally hosted software resources, including both site-licensed commercial software and public domain software. On site personnel support the following:

- Academic software (used primarily "in the classroom"): Ansys, Aspen, AutoCAD, B Jac, Control Station, Icarus, Magma 2.11, Maple (multiple versions), MathCAD, Mathematica, Matlab (multiple versions), Minitab, Ultimate Speed Reader, Working Model2d
- Microsoft Windows XP
- Utility Software: Adobe Acrobat, Corporate Time, Frontpage 2002, Leash, McAfee, Mediamatics Dvmatics, Office 2003, Office XP, Outlook, PDF995, Publisher 2002, Secure CRT, SecureFX, Visual Studio .NET, Visual Studio 6, Wake, WinRAR, XWin32

***Departmental computing resources:***

The CSSE departmental computer network is a subnet of the Rose-Hulman network. Computer science majors are frequently employed by the department as system managers and course assistants.

The department maintains a number of publicly available Sun workstations including 2 Ultra80s, 12 Ultra 10s, and 12 SunBlade 150s. There are an additional 12 Intel PC desktop computers, most of which are configured for dual boot (Windows and Linux). All faculty who want them have a Sun Ultra 5 or Ultra 10 in their offices, in addition to their laptop. There are additional machines that provide network and file services for the department's computers. Students help manage the entire departmental network as part of their non-academic education.

The department has two general purpose computing laboratories located in the center of the CSSE department office complex. The laboratories are configured so students can meet in small groups around one or two workstations and still have table space for papers, books, and their laptops. A third lab is dedicated for use in the Operating Systems and Computer Security courses. This lab has 12 computers that can be isolated from the department and campus networks for student projects.

The department provides a variety of software development tools, primarily through the Microsoft Software Development Network Academic Alliance (MSDNAA). The department's MSDNAA license includes Visio, Visual Studio .NET, Microsoft Project, and Microsoft Source Safe. The primary Java development environment is JCreator from Xinox Software, and the Concurrent Versions System (CVS) is the primary software used for configuration management.

***Other computing resources:***

The Imaging Systems Laboratory (an interdisciplinary laboratory jointly operated by the Departments of Computer Science and Software Engineering, Electrical and Computer

Engineering, Mathematics, and Physics and Optical Engineering) contains six SGI workstations and two Pentium-based workstations.

**Laboratory equipment planning, acquisition, and maintenance processes**

These topics are addressed in Section 7.[19]

## B. Student Access

All members of the campus community, including computer science majors, have unlimited access to the computing resources provided by the Institute. As described above, essentially all buildings on campus (in particular the residence halls, academic buildings, and student union) are wired for network access, and also have wireless access points. Computers in all classrooms, laboratories and faculty offices are connected to the campus network. All academic buildings and the Student Union also provide wireless networking. Off campus access is available via both dial-in and VPN.

All members of the campus community, including computer science majors, have unlimited 24-hour physical access to the department's general purpose laboratories and user-level access to its systems. Students enrolled in the operating systems and computer security courses have unlimited 24-hour physical access to the OS/Security laboratory to its systems.

Students enrolled in appropriate courses or engaged in appropriate project activity have unlimited 24-hour user-level access to the Imaging Systems Laboratory and its systems.

## C. Faculty Access

All Rose-Hulman faculty members receive a freshman laptop when they arrive, which may be used for class preparation, professional development, etc. These laptops are replaced on a three-year cycle. All faculty who want them have a Sun Ultra 5 or Ultra 10 in their offices, in addition to their laptop. The department's system administrator has root access to all of the department's machines, and faculty have unassisted root access to all laboratory machines.

## D. Support Personnel

Together with the Department of Mathematics, the Department of Computer Science and Software Engineering employs a full-time systems administrator to oversee the department's labs, equipment, and software, including installing, maintaining, and managing departmental hardware, software and networks. The systems administrator, in turn, supervises several students who work part-time to assist him with these duties.

The systems administrator is exceptionally capable, dedicated, and responsive. He is, however, spread very thin. During the coming academic year, the department head will request authority to hire a full time assistant system administrator.

The Office of Instructional, Administrative, and Information Technology (IAIT) provides campus-wide support for our internet connection and oversees the laptop program.

No faculty members provide significant hardware, network, or software support.

---

[19] The self-study questionnaire is self-contradictory, in that it specifically asks that this information not be duplicated here, and subsequently specifically asks for it.

## 7. Institutional Support and Financial Resources

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> The institution's support for the program and the financial resources available to the program are sufficient to attract and retain qualified faculty, administer the program effectively, acquire and maintain computing resources and laboratories, and otherwise provide an environment in which the program can achieve its educational objectives and outcomes. Support and resources are sufficient to provide assurance that the program will retain its strength throughout the period of accreditation.

### A. Faculty Stability

There are no significant problems attracting and retaining faculty. Faculty positions are typically advertised in *Communications of the ACM* and *IEEE Software*, both in print and online, as well as on the department's web site. The department looks for faculty who are committed to being excellent teachers, working with highly-motivated undergraduate students, and remaining professionally active. Applications are reviewed by the department faculty and screened for those candidates worthy of conducting a phone interview with. From these conversations, the pool is narrowed to those people who will be invited to campus for an onsite interview. On campus, the candidate meets with the department faculty, faculty from other departments the candidate is interested in talking to, students, the Dean of Faculty, and the President. The candidate is also given a tour of campus and the Terre Haute community. Most candidates leave campus with a very favorable impression of Rose-Hulman and its environment, and look forward to receiving an offer from the school to join its faculty. Rose-Hulman offers its faculty a very competitive salary, an excellent benefits package (including institute contributions to TIAA/CREF retirement plans at 10% of salary, health insurance, life insurance, and disability insurance), an opportunity to apply for a sabbatical leave every seven years, and institute funding to attend at least one professional conference each year. The institute, and the department in particular, offer an environment where everyone is respected, where everyone supports each other, and where interactions between students, faculty, and staff are encouraged and enjoyed. The stability of the faculty is illustrated quantitatively by Table 16.

**Table 16. Total number of full-time faculty, resignations, retirements, and new hires for each of the last five years.**

| Year | Total Faculty | Resignations | Nonrenewals | Retirements | New Hires |
|---|---|---|---|---|---|
| 2001-2002 | 9 | 0 | 0 | 0 | 0 |
| 2002-2003 | 10 | 0 | 0 | 1 | 2 |
| 2003-2004 | 11 | 1 | 0 | 1 | 3 |
| 2004-2005 | 11 | 0 | 0 | 0 | 0 |
| 2005-2006 | 13 | 1 | 0 | 0 | 3 |

### B. Faculty Professional Activities

All faculty members are actively engaged in professional development. They attend both technical conferences and pedagogical conferences. In the last academic year, the department faculty attended 30 different conferences and workshops (an average of 2.7 per person), and authored 20 different papers (an average of 1.8 per person). Department faculty regularly serve on

conference committees, are nominated for several editions of *Who's Who*, and have achieved Senior Member status in professional organizations.

### C. Administration Effectiveness

There is a relatively flat administrative structure at Rose-Hulman. The department head reports to the Dean of Faculty, who reports to the President. The department head meets regularly with the Dean (at least once every two weeks) to discuss all matters pertinent to the department. There is a very open and mutually respectful relationship between the department head and the Dean. Each summer, the department head and the Dean meet with the President to discuss the department in general. Communication of all issues between the Dean, the department head, and the department faculty and staff is considered extremely important by all concerned, and every effort is made to insure the communication channels remain open.

### D. Adequacy of Resources

All academic departments at Rose-Hulman are organized in a single administrative structure. Consequently there is not a "college of engineering" that must compete with other colleges within a university structure. Because 85% of our students major in engineering and 8% in computer science, it is clear that adequate support of engineering programs is inextricably tied to the continued vigor of the institution.

Rose-Hulman is in sound financial condition as is indicated by an "A1" rating from Moodys Investors Service. The Institute's operating budget is approximately $70 million per year. Approximately 65% of revenue comes from student tuition. The discount rate is 31%. The Institute currently draws $13,885,000 yearly from its endowment of approximately $160 million.

The Institute's previous 10-year capital campaign which ended in 2004 resulted in $250 million. A new campaign is expected to be launched within two years. Compensation goals described in our 2000 ABET report (faculty salaries at the 60th percentile of the Association of Independent Technological Universities (AITU), professional salaries at the 60th percentile of institutions of similar size as reported by College and University Personnel Association (CUPA), and average hourly staff wages at the 80th percentile of local markets) have become dated. Consequently the Board of Trustees has recently formed a Commission for comprehensive review of the Institute salary structures.

The Institute maintains professional development funds adequate to allow each faculty member to attend at least one national meeting per year. One-third of this support comes from the dean of faculty's budget. The Institute maintains an active sabbatical program that provides one full academic-year release at half pay (or 2 quarters release at 4/5 pay, or one quarter at full pay) every seven years.

Rose-Hulman Institute begins its budget cycle early in the academic year. In the fall, all units begin thinking about their budget needs. Each academic department submits budget requests to the Vice President for Academic Affairs. The Vice President for Academic Affairs meets with each department head and may have discussions with the department heads as a whole to help prioritize expenditures and to find possible partnerships which create not only a better use of resources but also a better shared experience.

Typically, first requests for the budget from academic departments are solicited by the Vice President for Academic Affairs (VPAA) in November. After suggesting modifications, the VPAA communicates all these to the Vice President and Chief Administrative Officer (CAO). The CAO prepares a first draft of the budget for presentation to the President. Then a meeting is called that includes all the vice presidents, department heads, and other key section heads. Through a series of these meetings and further budget revisions, a final budget is crafted. The process is transparent and collaborative, which helps us to express our needs, discuss competing

interests, and eventually meet our budget goals. The President submits the final budget to the Executive Committee of the Board of Trustees in late January. They discuss with the President any questions about the proposed budget. The Vice President for Academic Affairs is present in these meetings. Once the Executive Committee has made any suggestions, the President and Vice Presidents consider these suggestions and make any necessary changes. The finalized budget is then submitted to the Board of Trustees at its annual winter meeting, held in late February. The budget includes all elements, ranging from salary to equipment. Once the budget is approved at the February Board Meeting, letters of appointment for the following year are issued to current faculty members.

In the next three sub-sections we describe three major institutional-level developments that have had a major impact during the six years since the previous ABET visit. The three developments in instructional equipment, computer infrastructure, and project space indicate the institutional commitment to constant improvement of our engineering programs.

Resources for the department are very adequate. The department has sufficient faculty, equipment, and space to adequately carry out its teaching mission, and there is adequate space and equipment for faculty to carry out their professional obligations. At the institution level, all the academic department heads and the Dean of Faculty meet once each week as the academic administrative team. These meetings build and reinforce the collegiality at Rose-Hulman, making it easy for all departments to work with each other in supporting the needs of each and every student and academic program at Rose-Hulman.

### E. Laboratory and Computing Resources

Rose-Hulman faculty are encouraged to write grants for new equipment and laboratory facilities. There is institutional support through the Development Office and Business Office to help faculty be successful in these endeavors. A few years ago the department heads convinced the Dean that a line item for replacing laboratory equipment at the end of the equipment's life span needed to be included in the institute's budgeting process. The Dean asked the department heads to take inventory of all department equipment, including replacement cost and expected life span. The individual lists were collated, and it was determined that the institute needed to invest $1M per year to replace aging equipment. The Board of Trustees approved this budgeting guideline, and the institute has been ramping up this equipment line item by adding $200K annually to the budget until the $1M level is reached (during 2005-6 it was $600K, and for 2006-7 it will be $800K). Each fall, the department heads prepare a prioritized list of equipment needs for their departments, which the Dean collates and distributes to the department heads. At one of the department heads meetings, the group discusses the collated list and prioritizes the needs at an institute level. This is all done in a very collegial and cooperative manner, insuring each department gets its fair share of the equipment funds.

### F. Continuity of Institutional Support

As the department has grown, in terms of student enrollments and course requirements for the major, the administration has allowed the department to hire additional faculty to meet the needs of the department. The department has also been able to hire a full-time systems administrator to oversee the department's laboratories and computer network, insuring a faculty member would not have to do this work. Three years ago the institute invested just under $2M to renovate space that became available in the main academic building (Moench Hall) for a new home for the department. This renovation provided office space for all faculty and staff to be together in one physical area, two very spacious general purpose laboratory rooms, and one smaller laboratory for operating systems and computer security classes. The Dean continues to be a strong supporter of the department and the direction in which it is heading; thus no problems are anticipated in this area for the foreseeable future.

## 8. Institutional Facilities

The purpose of this section is to provide evidence that the computer science program satisfies the following criterion:

> Institutional facilities including the library, other electronic information retrieval systems, computer networks, classrooms, and offices are adequate to support the educational objectives and outcomes of the program.

### A. Library Staffing

The Institute has a centralized library/information system. Staffing consists of 3 FTE librarians, 1 FTE support library worker, and 8 FTE student workers. This number of professional librarians and support personnel is adequate to support the computer science program.

### B. Library Technical Collection

The library strives chiefly to model electronic full text technical information available 24/7. In support of the CS curriculum the library manages electronic subscriptions to the ACM Digital Library, IEEE ASPP package, EBSCO's Academic Search Premier and Business Search Premier (offering 10,000+ subscription, both peer-reviewed and trade), and such reference and text tools as Safari Tech Books Online and Knovel.

### C. Library Electronic Access

In addition to the items mentioned above which function both as sources of full text and bibliographic indexes, the library manages a number of finding tools to assist faculty and students in identifying peer-reviewed materials. The chief tools include Engineering Village (Ei-Compendex – Engineering Index, 1969-to date) and ScienceDirect (College Edition).

### D. Classroom Equipment

All classrooms at Rose-Hulman are under control of the Registrar's Office, and no one classroom is assigned to any one department. Classrooms that computer science courses are typically taught in contain 24 to 40 seats, multiple whiteboards, power and network ports for students' laptops, a faculty desk which also contains power and network ports, a ceiling projector for displaying a laptop screen, and videotape and DVD playback capabilities. Most classrooms have moveable chairs, so students can get into small teams and do team project work during a class session. The rooms computer science courses have been assigned to have proven adequate for teaching the courses. All of this is made more flexible because each student owns a laptop, and brings it to class when necessary for our in-class labs. We additionally have licenses for students to load most required software onto those laptops. Thus, we don't in general need special rooms with software loaded on certain machines there.

In the summer of 2001 the dean of faculty requested all departments to perform a comprehensive inventory of instructional equipment. The inventory included date of purchase, replacement costs, anticipated lifetime of the item if replaced, and an evaluation of state of current terms, rating each as "state-of-the-art," "modern," "adequate," "obsolete," or "embarrassing."

The survey indicated two things: (1) only 37% of the instructional equipment was rated as state-of-the-art, modern or adequate and (2) that the Institute was under-budgeting instructional equipment by over $1 million per year just to maintain steady state.

As a result, the Institute leadership committed to a strategic goal of increasing the yearly operating budget for instructional equipment by $200,000 per year for each of 5 years in order to

eliminate the $1 million per year short fall. As a consequence, the FY 2006-07 budget includes $800,000 more for instructional equipment than did the FY 2002-03 budget.

The summer instructional equipment survey is now a permanent part of our equipment replacement process. Figure 7.1 shows the systematic improvement in instructional equipment at the Institute. The summer 2005 inventory indicated that nearly 78% of the Institute's instructional equipment is rated by departments as state-of-the-art, modern, or adequate.

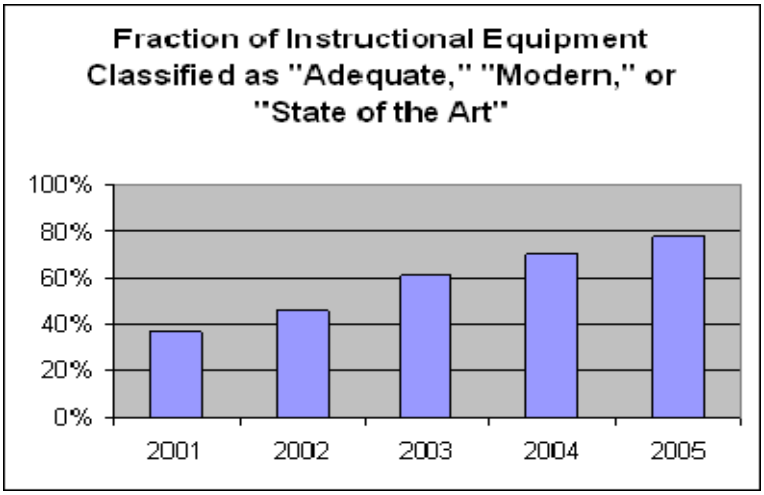



**Figure 5. Improvement in the quality of instructional equipment at Rose-Hulman since commitment to reach $1,000,000 yearly allocation.**

It should be noted that the increased funds are institutionally budgeted to the dean of faculty, not individual departments. Individual departments make requests each year. These requests are compiled by the dean and distributed to department heads. The dean and assembled department heads then meet to discuss one another's respective needs. In a collegial and Institute-oriented discussion, department heads formulate by consensus their recommendation to the dean for the yearly disbursal of funds.

In 1999, the Myers Center for Technological Research with Industry was dedicated. This 40,000 square foot building is used for student project space, with priority given to projects being done for external clients (e.g. senior engineering design projects). Individual departments are not allowed to "own" space in the building. Rather, space is assigned to project teams on an "as needed" basis. Yearly space allocation is overseen by the associate dean of the faculty. This flexible allocation allows for optimum space utilization as various program enrollments rise and fall.

In addition to flexible project space for both design and construction, the Myers Center houses several support facilities used by students in all departments: a machine shop, rapid prototyping machines (starch and ABS plastic), computer-controlled MTS machine, microscopy lab (x-ray, SEM, fluorescence, optical), and chemical analysis. Here again our guiding philosophy is that of shared resources for maximum utilization.

In addition, it should be noted that four of the under-utilized high bays in the Myers Center have been "capped" providing a second floor in this area. This increased space has been used to provide laboratory/classrooms for the rapidly growing Biomedical Engineering program. When funding is obtained for the planned Biomedical Engineering building, the created space in the Myers Center will become additional project space.

## E. Faculty Offices

All faculty in the Department of Computer Science and Software Engineering are given an office of sufficient size to conduct their professional obligations. A team of three or four students can easily meet with a faculty member in the faculty member's office. Each office is equipped with sufficient desk space, chairs, bookshelves, filing cabinets, and whiteboard. A docking station, external monitor, and external keyboard are provided for connecting the faculty member's laptop to. In addition, a workstation of the faculty member's choosing is also provided. Wired and wireless network access is provided in each office, as is a telephone with voice mail capability.

# Appendix I. Information Relative to the Entire Institution

## A. General Information

| | |
|---|---|
| Institution | Rose-Hulman Institute of Technology |
| Department | Computer Science and Software Engineering |
| Street | 5500 Wabash Ave |
| City | Terre Haute |
| State | Indiana |
| Zip | 47803 |
| URL | http://www.rose-hulman.edu |

Name and Title of Chief Executive Officer of Campus (President, Chancellor, etc.)

| | |
|---|---|
| Robert E. Bright | Chief Executive Officer, AY 2005-2006 |
| Gerald Jakubowski | President, beginning AY 2006-2007 |
| (Name) | (Title) |

## B. Type of Control

| | |
|---|---|
| Private, non-profit | X |
| Private, other | |
| Federal | |
| State | |
| Municipal | |
| Other (specify) | |
| Affiliation, if private | None |

## C. Regional or Institutional Accreditation

Rose-Hulman is accredited by the North Central Association of Colleges and Schools. A copy of the most recent accreditation action is shown in Figures.

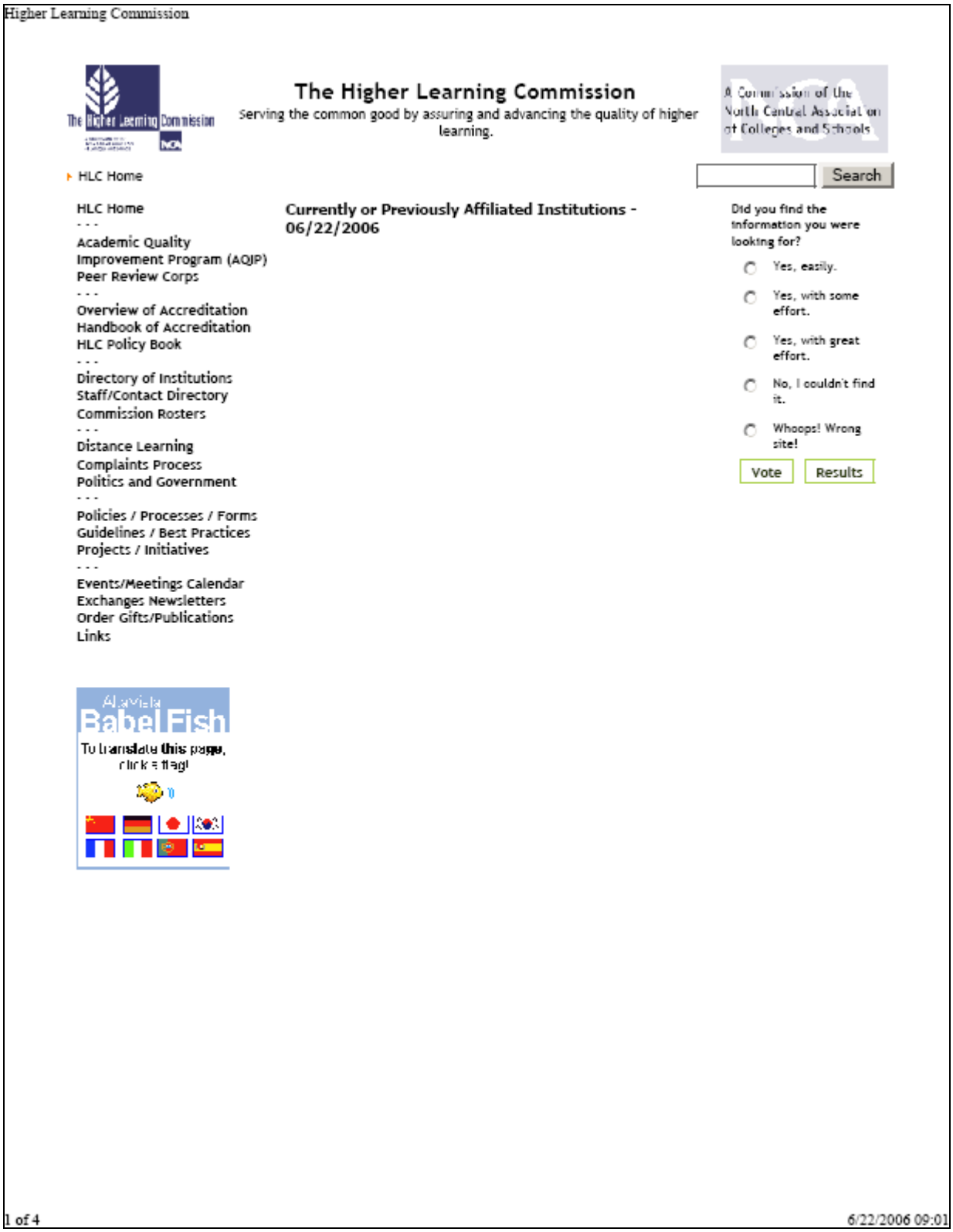



**Figure 6. Most recent accreditation action of North Central Association of Colleges and Schools (page 1 of 4).**

-- Back to Institution Directory --

# Rose-Hulman Institute of Technology

5500 Wabash Ave.
Terre Haute, IN 47803-3999

http://www.rose-hulman.edu

*Chief Executive Officer:* Mr. Robert Bright, Chairman of the Board

*Name change notes: Rose Polytechnic Institute to Rose-Hulman Institute of Technology (1970)*

*Current Accreditation Status:* Accredited

*Accreditation Date(s):* (1916- .)
*Commission Participation:* AQIP PARTICIPANT

*Year of Admission to AQIP:* 11/20/2000
*Year of Next Reaffirmation of Accreditation:* 2007 - 2008
*Year of Next System Appraisal:* 2008 - 2009
*Due Date of Next Systems Portfolio:* 6/1/09
*Last Action:* 08/07/1992

*Legal Status:* Private NFP
*Degrees Awarded (details below):* B, M

---

*Stipulations on Affiliation Status:*

None.

*Approval of New Degree Sites:*

Prior Commission approval required.

*Approval of Distance Education Degrees:*

Prior Commission approval required.

*Reports Required:*

None.

*Other Visits Scheduled:*

None.

---

Enrollment Headcount (last updated: 06/09/2006)

| | Full-Time | Part-Time |
|---|---|---|
| Undergraduate: | 1765 | 3 |
| Graduate: | 38 | 84 |



**Figure 7. Most recent accreditation action of North Central Association of Colleges and Schools (page 2 of 4).**

| | | |
|---|---|---|
| Post-baccalaureate First Professional: | 0 | 0 |

Other Headcounts (last updated: 06/09/2006)

| | |
|---|---|
| Non-Credit headcount: | 7 |
| Dual enrollment (high school) programs: | 2 |

Degree Programs (last updated: 06/09/2006)

| | Programs Offered | Degrees Awarded in Last Reported Year |
|---|---|---|
| Associate Degrees | 0 | 0 |
| Bachelors Degrees | 15 | 368 |
| Masters Degrees | 9 | 38 |
| Specialist Degrees | 0 | 0 |
| First Professional Degrees | 0 | 0 |
| Doctoral Degrees | 0 | 0 |

Certificate Programs (last updated: 06/09/2006)

| | Programs Offered | Certificates Awarded in Last Reported Year |
|---|---|---|
| Pre-Associate Certificates | 0 | 0 |
| Other Undergraduate Certificate | 5 | 0 |
| Post Baccalaureate Certificates | 0 | 0 |

Off-Campus Activities (last updated: 06/09/2006) Click here for definitions...

| | | |
|---|---|---|
| In-State: | Campuses: | None |
| | Sites: | Bloomington (Smith and Third) ; Crane (Crane Naval Surface Warfare |



**Figure 8. Most recent accreditation action of North Central Association of Colleges and Schools (page 3 of 4).**

| | | |
|---|---|---|
| | | Center) ; Indianapolis (Environmental Management Inc.) ; Indianapolis (Knowledge Services) |
| | Course Locations: | None |
| Out-of-State: | Campuses: | None |
| | Sites: | None |
| | Course Locations: | None |
| Out-of-U.S.: | Campuses: | None |
| | Sites: | None |
| | Course Locations: | None |

Distance Learning (last updated: 06/09/2006)

None

-= Contents © 2006 Higher Learning Commission -- Site powered by Mambo Open Source =-



**Figure 9. Most recent accreditation action of North Central Association of Colleges and Schools (page 4 of 4).**

## D. Enrollment[20]

| | |
|---|---|
| Total enrollment for the entire institution (FTE) | 1766 full-time undergraduate students<br>9 part-time undergraduate students<br>1927 FTE undergraduate students<br>38 full-time graduate students<br>74 part-time graduate students<br>66 FTE graduate students |
| Total faculty for the entire institution (FTE) | 142 full-time tenure track<br>2 part-time tenure track<br>6 full-time non-tenure track<br>5 full-time non-tenure track |

## E. Promotion and Faculty Tenure

Promotion and tenure are based on three components: teaching, professional development, and service. New faculty hires are given a six-year probationary period. The department head writes an annual performance review which is shared with the faculty member and put in the faculty member's PTR (Promotion, Tenure, and Retention) folder in the Dean of the Faculty's office. At

---

[20] Source: Office of Institutional Research, Planning and Assessment

Definitions:

* Whenever possible, Common Data Set definitions are used.

Instructional Faculty is defined as those members of the instructional-research staff whose major regular assignment is instruction, including those with released time for research.

Full-time instructional faculty: faculty employed on a full-time basis for instruction (including those with released time for research)

Part-time instructional faculty: Adjuncts and other instructors being paid solely for part-time classroom instruction. Also includes full-time faculty teaching less than two semesters, three quarters, two trimesters, or two four-month sessions. Employees who are not considered full-time instructional faculty but who teach one or more non-clinical credit courses may be counted as part-time faculty.

Full-time equivalent (FTE) instructional faculty (full-time plus 1/3 part-time).

Student numbers are as of RHIT's official fall reporting date (17th day of classes) and include non-degree seeking students.

For students, 1 FTE equals 15 credit-hours per term of institutional undergraduate course work, meaning all courses--engineering, humanities and social sciences, etc., and 12-credit hours per term of institutional graduate course work.

the end of the third year, the faculty member's performance is given a formal review by the Institute's PTR Committee, the Dean of the Faculty, and the department head. It provides an opportunity for the PTR Committee and the Dean to let a faculty member know his or her strengths and weaknesses at the time, giving ample opportunity for adjustment before a tenure decision is made. In a faculty member's sixth year of service a formal tenure decision is reached (this decision may come earlier for faculty with prior teaching experience). Faculty who are not granted tenure are given a terminal contract for the following year, giving them opportunity to find a position elsewhere. Promotion from assistant to associate professor usually occurs at the same time a tenure decision is made. Promotion to full professor is considered no earlier than six years following promotion to associate professor.

Within the Department of Computer Science and Software Engineering, the department head discusses with the tenured full professors in the department recommendations for promotion and tenure of the other faculty members. When a consensus is reached to recommend promotion and/or tenure for a colleague, the department head submits the recommendation to the Dean of the Faculty and the PTR Committee. The faculty member is required to submit a portfolio describing his or her teaching, professional development, and service to the PTR Committee. The committee meets with the department head and the Dean of the Faculty to discuss the promotion and/or tenure decision. The committee makes its recommendation to the President of the Institute; the Dean likewise submits his recommendation to the President.

Each year the President of the Institute announces what the average salary adjustment will be for the following year when the budgeting process begins. The Dean gives each department head a pool of money for salary adjustments which reflects this average increase for their department's salaries. The department head is free to allocate this salary increase as he or she sees fit, awarding those faculty members who excelled higher than average salary increases and those whose performance is not as strong smaller increases.

# Appendix II. General Information on the Unit Responsible for the Program

**A. Program Unit**

| |
|---|
| Department of Computer Science and Software Engineering |
| http://www.cs.rose-hulman.edu |

**B. Administrative Head of Program Unit**

| | |
|---|---|
| Cary Laxer | Head, Department of Computer Science and Software Engineering |

## C. Organization Chart

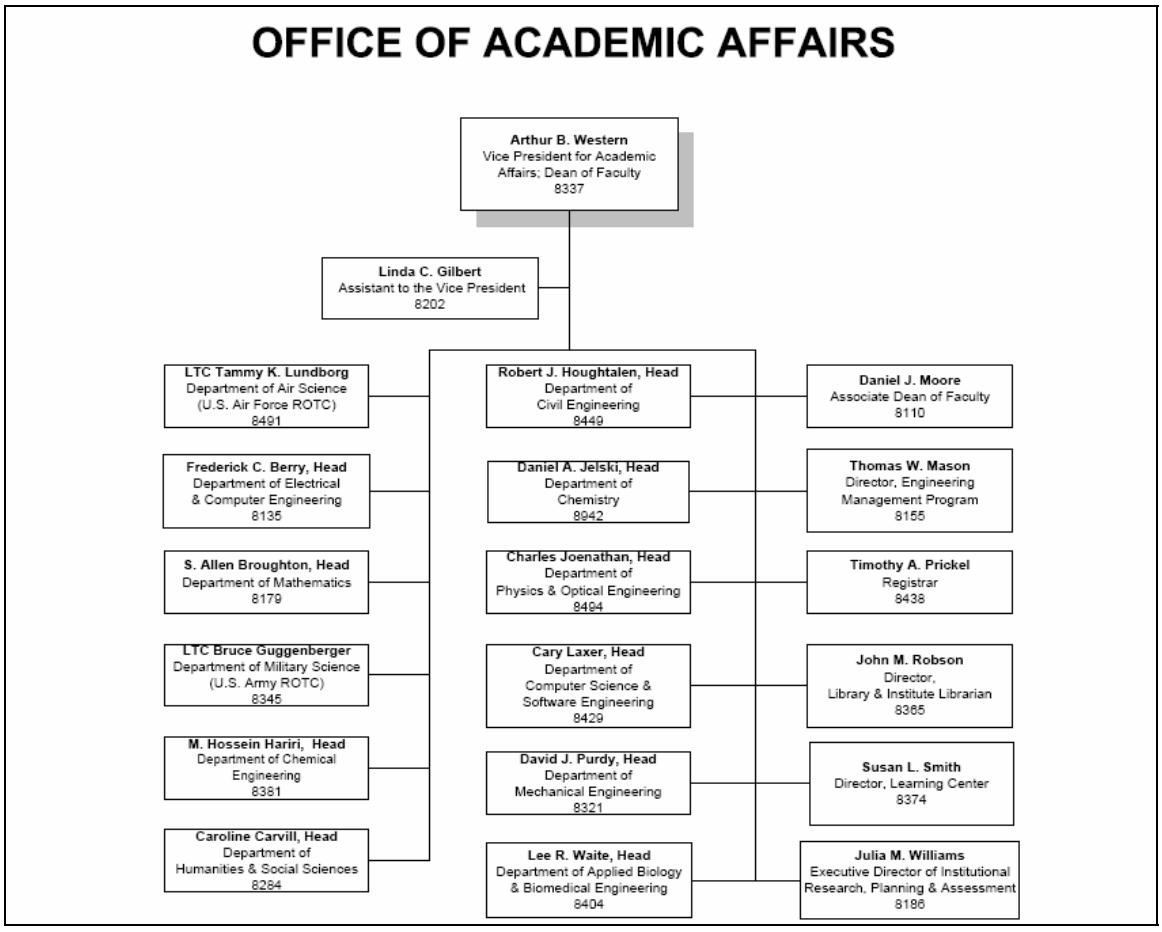



**Figure 10. Organizational chart illustrating direct supervisory relationship between Dean of Academic Affairs and department heads.**

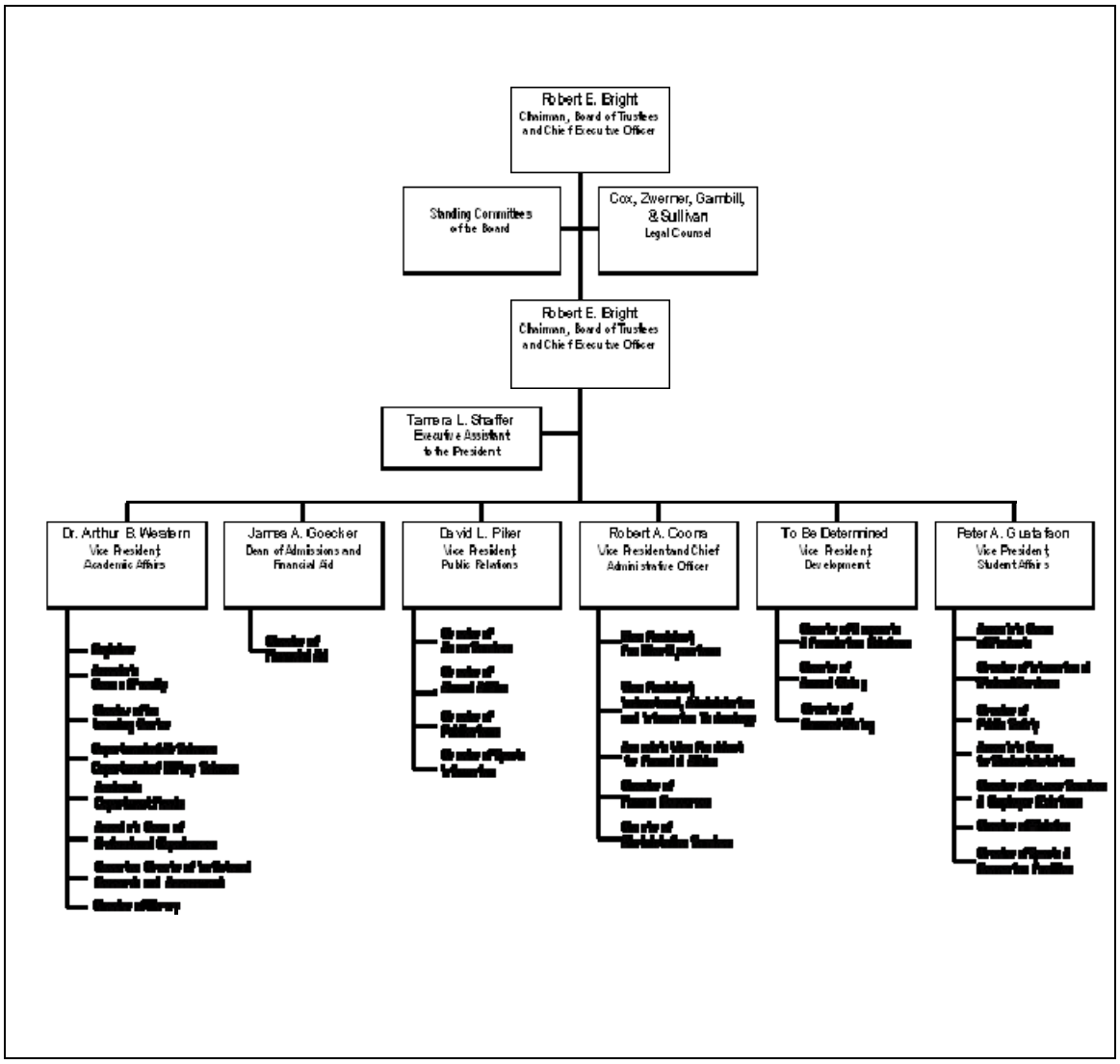



**Figure 11. Organizational chart illustrating direct supervisory relationship between President's position and Dean of Academic Affairs.**

## D. Computer-Related Undergraduate Degree Programs

The three computer related degree programs indicated in Table 17 have separate listings in the undergraduate catalog under the "Programs of Study" section. Each listing describes the respective academic program, lists the goals and objectives of the program, provides a suggested program of study that would lead to the degree, and specifies the requirements for the degree. Each degree program also has its own web page, from which further information can be gleaned.

During the fall term of the freshman year, all students are required to take a course called College and Life Skills. One of the requirements of this course is to attend an evening academic presentation session, where faculty members from each academic major at Rose-Hulman discuss the major and career opportunities in it. Students are given plenty of time to ask questions. On purpose, the presentations for computer science, computer engineering, and software engineering are held in the same room (a large lecture hall, as approximately 120 students may attend; electrical engineering is also included in this group). Many students know they would like to study one (or more) of these disciplines, but are unsure which one is right for them. This gives

them the opportunity to listen to all of them, compare and contrast them, and ask questions that may be answered by more than one faculty member.

**Table 17. Computer-related undergraduate degree programs offered by Rose-Hulman.**

| Program Title | Years Required | Degree Awarded | Administrative Unit | If accredited, by whom |
|---|---|---|---|---|
| Computer Science | 4 | Bachelor of Science | Department of Computer Science and Software Engineering (CSSE) | |
| Computer Engineering | 4 | Bachelor of Science | Department of Electrical and Computer Engineering (ECE) | ABET EAC |
| Software Engineering | 4 | Bachelor of Science | Department of Computer Science and Software Engineering (CSSE) | |

# Appendix III. Finances

## A. Finances Related to the Program(s)

**Table 18. Financed related to the Computer Science program.**

| | Institutional Funds | Non-recurring or Outside Funds |
|---|---|---|
| Administrative Salaries | 118,189 | 0 |
| Faculty Salaries | 866,722 | 11,403 |
| Non-teaching Professionals' Salaries[2] | 0 | 0 |
| Support Personnel Salaries & Wages | | |
| Secretarial | 30,786 | 0 |
| Technician | 62,019 | 0 |
| Other (specify) | 2,907 | 27,578 |
| Graduate Students | | |
| Operating Expenditures (Excluding research operations and travel) | 19,939 | 10,332 |
| Capital Equipment Expenditure: | | |
| Teaching | 28,400 | 4,000 |
| Research | 0 est | 0 est |
| Computer Expenditures: (total, including value of allocated computer time from other sources for teaching and research) | | |
| Hardware | 2,248 est | 0 est |
| Software | 2917 est | 0 est |
| Allocated time | Data not available | Data not available |
| Travel Expenditures (non-research funds) | 24,632 | 5,012 |
| Scholarship Awards (if administered by the program unit) | N/A | N/A |
| Library (if administered by program unit) | N/A | N/A |
| Research (if separately budgeted) | N/A | N/A |
| Other (specify) Fringes | 390,781 | 1,151 |
| Total | 1,504,632 | 55,475 |

## B. Operating and Computing Expenditures for the Five Fiscal Years Immediately Preceding that Reported in III A

**Table 19. Operating expenses for the CSSE department.**

| Fiscal Year | 03 | 04 | 05 | 06 (Budget) | 07 (Budget) |
|---|---|---|---|---|---|
| Institutional Funds | 10,102 | 18,014 | 19,939 | 22,900 | 26,275 |
| Outside Funds | 6,372 | 3,217 | 10,332 | # | # |

# - Because they cross fiscal years, individual grants are not included in the annual budgets.

**Table 20. Computer hardware/software capital expenditures (excluding equipment used primarily for research) for the CSSE department.**

| Fiscal Year | 03 | 04 | 05 | 06 (Budget) | 07 (Budget) |
|---|---|---|---|---|---|
| Institutional Funds | | | | | |
| Outside Funds | | | | | |

## C. Additional Funding

Professional development travel is supported at the Institute level by providing 50% matching funds from the Dean of the Faculty's budget. Thus, a faculty member's professional development travel is funded two-thirds from the department budget and one-third from the Dean's budget. This support is available to all faculty members at the Institute. For the Department of Computer Science and Software Engineering this adds approximately $15K to our budget for professional travel. Additional professional development support for computer science and software engineering faculty comes from the department's restricted account, an account that receives gifts from alumni and other supporters. This amounts to approximately $10K per year.

# Appendix IV. Program Personnel and Policies Towards Consulting, Professional Development, and Recruiting

## A. Term of Appointment of Administrative Head

9 month _____ 12 Month X Other (specify) _____

## B. Number of Personnel Associated with Program

**Table 21. Number of personnel associated with the Computer Science program.**

| | Full-time | Part Time | | Total |
|---|---|---|---|---|
| | Number | Number | FTE | FTE |
| Faculty | 12 | 0 | 0 | 12.00 |
| Non-teaching Professionals | 0 | 0 | 0 | 0.00 |
| Administrative head | 1 | 0 | 0 | 1.00 |
| Administrative | 0 | 0 | 0 | 0.00 |
| Computer Lab Personnel: | | | | |
| Professionals | 0 | 0 | 0 | 0.00 |
| Technicians | 1 | 0 | 0 | 1.00 |
| Secretarial, Accounting, etc. | 1 | 0 | 0 | 1.00 |
| Graduate Teaching Assistants | 0 | 0 | 0 | 0.00 |
| Graduate Research Assistants | 0 | 0 | 0 | 0.00 |
| Graduate Students | 0 | 0 | 0 | 0.00 |
| Undergraduate Students | 202 | 0 | 0 | 219.67 |

## C. Policies

This section describes institutional and departmental policies regarding faculty consulting and other professional development, faculty workload, and faculty recruiting.

### Private Consulting, Sponsored Research, and Extra Compensation

Computer science and software engineering faculty are encouraged to be actively involved in private consulting and sponsored research as two ways to remain professionally active. Faculty are entitled to 20% of their time for consulting and other professional development activities. Income faculty earn from private consulting is theirs to keep, and does not need to be reported to Rose-Hulman. Research grant funding can buy out a part of a faculty member's time, with the funds used to hire additional faculty (visiting or adjunct) to cover the reduced loads for faculty on sponsored research for the duration of the grant.

**Faculty Workload**

Faculty in computer science and software engineering are normally expected to teach 12 contact hours per term, with one or two preparations. The department head is granted two-thirds release time each term for administration of the department, and thus only teaches 4 contact hours per term. One faculty member is designated the Director of Software Engineering and received one-third release time each term, teaching eight contact hours per term. One member of the department faculty serves as the Director of the Imaging Systems Laboratory and receives a one-course release during one term for administering this lab. Each faculty member must remain professionally active, and should devote about 20% of his or her time to this endeavor. In addition, each faculty member serves as an academic advisor and has institute committee work, which should take up about 5-10% of that person's time.

**Faculty Recruitment**

The department faculty determines additional faculty needs by carefully reviewing current staffing levels, trends in enrollment, and changing needs of the curriculum. A case is made to the Dean of the Faculty, and upon his approval, advertising for the position is initiated. The department usually places print and online ads in some combination of Communications of the ACM, IEEE Software, and IEEE Computer. A notice is also placed on the department's website. Interested applicants are requested to submit a letter of application, a current resume, statements on the teaching and professional development philosophies, and at least three letters of recommendation. Department faculty review the applications and categorize them as high interest, average interest, and low interest. Those in the high interest category are given an initial phone interview with two to four department faculty that lasts approximately one hour. This interview is used to gauge the interest and qualifications of the candidate, and to make sure the candidate is aware of the nature of our department and our institution. The department faculty as a whole review the comments of the interviewers after the interviews are completed, with the goal of reducing the candidate pool to three or four people that will be invited to campus for an onsite interview. These candidates are discussed with the Dean of the Faculty for a final decision before inviting them to campus. On campus, candidates have a social hour with department faculty, staff, and spouses the night before the interview. During the interview, candidates meet one on one with each faculty member in the department, the Dean of the Faculty, and the President. Members of the department's Student Advisory Committee have breakfast with the candidates. The candidates are welcome to ask to meet with faculty in other departments on campus that they may have an interest in. Candidates give a one hour seminar to the campus community. Students majoring in computer science and software engineering are invited to stay after the seminar to talk to the candidate about their presentation, the department, as well as the candidate themselves. During this meeting, no faculty are present. Following all candidates' interviews, the department faculty take their input and the students' input and try to reach a consensus on who to offer a position to. The department head carries this recommendation to the Dean of the Faculty. If the Dean concurs, terms of an offer are discussed between the department head and the Dean, and the department head notifies the candidate. If the Dean does not concur, the department head brings the Dean's concerns back to the department faculty for further discussion.

There are no barriers to hiring the appropriate faculty. In fact, Rose-Hulman has been known to make a position available if a "superstar" comes along that wants to be part of our campus and that Rose-Hulman wants to have.

# Appendix V. Program Enrollment and Degree Data[21]

**Table 22. Five year history of enrollment and degrees conferred (Rose-Hulman as a whole).**

| | | | Enrollment Year | | | | | | Total | Total | Degrees Conferred | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | AY | FT/PT | 1st | 2nd | 3rd | 4th | 5th | T | UG | Grad | BS | MS | PhD | Other |
| | 2005-06 (Current) | FT | 449 | 458 | 425 | 369 | 43 | 21 | 1765 | 38 | 0 | 0 | NA | 0 |
| | | PT | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 74 | 0 | 0 | NA | 0 |
| 1 | 2004-05 | FT | 473 | 468 | 395 | 346 | 47 | 27 | 1756 | 46 | 368 | 37 | NA | 1 |
| | | PT | 0 | 0 | 0 | 2 | 2 | 0 | 4 | 92 | 0 | 0 | NA | NA |
| 2 | 2003-04 | FT | 491 | 428 | 368 | 366 | 33 | 22 | 1708 | 54 | 346 | 43 | NA | NA |
| | | PT | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 88 | 0 | 0 | NA | NA |
| 3 | 2002-03 | FT | 451 | 394 | 391 | 332 | 54 | 13 | 1635 | 51 | 372 | 48 | NA | NA |
| | | PT | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 111 | 0 | 0 | NA | NA |
| 4 | 2001-02 | FT | 403 | 413 | 326 | 339 | 47 | 15 | 1543 | 57 | 359 | 34 | NA | NA |
| | | PT | 1 | 1 | 11 | 14 | 1 | 0 | 28 | 116 | 0 | 0 | NA | NA |
| 5 | 2000-01 | FT | 436 | 377 | 368 | 316 | 52 | 17 | 1566 | 58 | 327 | 33 | NA | NA |
| | | PT | 0 | 1 | 3 | 2 | 2 | 0 | 8 | 86 | 0 | 0 | NA | NA |

[21] Official fall term enrollment figures (head count) for the current and preceding five academic years and undergraduate and graduate degrees conferred during each of those years.

The "current" year means the academic year preceding the fall visit.

Official enrollment numbers reflect students' primary major designation as recorded by the Registrar's office.

Degrees conferred reflect students' primary major designation as recorded by the Registrar's office.

**Table 23. Five year history of enrollment and degrees conferred (CSSE Department).**

| | | | Enrollment Year | | | | | | Total | Total | Degrees Conferred | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | AY | FT/PT | 1st | 2nd | 3rd | 4th | 5th | T | UG | Grad | BS | MS | PhD | Other |
| | 2005-06 | FT | 42 | 57 | 47 | 50 | 3 | 3 | 202 | N/A | 0 | N/A | N/A | N/A |
| | (current) | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | N/A | N/A | N/A |
| 1 | 2004-05 | FT | 55 | 45 | 56 | 36 | 3 | 4 | 199 | N/A | 40 | N/A | N/A | N/A |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | N/A | N/A | N/A |
| 2 | 2003-04 | FT | 48 | 61 | 43 | 51 | 3 | 4 | 210 | N/A | 55 | N/A | N/A | N/A |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | N/A | N/A | N/A |
| 3 | 2002-03 | FT | 60 | 46 | 56 | 51 | 7 | 0 | 220 | N/A | 56 | N/A | N/A | N/A |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | N/A | N/A | N/A |
| 4 | 2001-02 | FT | 53 | 61 | 52 | 49 | 6 | 3 | 224 | N/A | 52 | N/A | N/A | N/A |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | N/A | 0 | N/A | N/A | N/A |
| 5 | 2000-01 | FT | 60 | 61 | 54 | 39 | 5 | 3 | 222 | N/A | 37 | N/A | N/A | N/A |
| | | PT | 0 | 0 | 1 | 0 | 0 | 0 | 1 | N/A | 0 | N/A | N/A | N/A |

**Table 24. Five year history of enrollment and degrees conferred (Computer Science program).**

| | | | Enrollment Year | | | | | | Total | Total | Degrees Conferred | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Year | AY | FT/PT | 1st | 2nd | 3rd | 4th | 5th | T | UG | Grad | BS | MS | PhD | Other |
| | 2005-06 | FT | 27 | 39 | 38 | 40 | 3 | 0 | 147 | NA | 0 | NA | NA | NA |
| | (Current) | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NA | 0 | NA | NA | NA |
| 1 | 2004-05 | FT | 35 | 38 | 45 | 30 | 3 | 2 | 153 | NA | 35 | NA | NA | NA |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NA | 0 | NA | NA | NA |
| 2 | 2003-04 | FT | 44 | 60 | 38 | 51 | 3 | 4 | 200 | NA | 55 | NA | NA | NA |
| | | PT | 0 | 0 | 0 | 1 | 0 | 0 | 1 | NA | 0 | NA | NA | NA |
| 3 | 2002-03 | FT | 60 | 46 | 56 | 51 | 7 | 0 | 220 | NA | 56 | NA | NA | NA |
| | | PT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NA | 0 | NA | NA | NA |
| 4 | 2001-02 | FT | 53 | 61 | 52 | 49 | 6 | 3 | 224 | NA | 52 | NA | NA | NA |
| | | PT | 0 | 0 | 2 | 2 | 0 | 0 | 4 | NA | 0 | NA | NA | NA |
| 5 | 2000-01 | FT | 60 | 61 | 54 | 39 | 5 | 3 | 222 | NA | 37 | NA | NA | NA |
| | | PT | 0 | 0 | 1 | 0 | 0 | 0 | 1 | NA | 0 | NA | NA | NA |

# Appendix VI. Admission Requirements

## A. Admission of Students

### Admissions

Students are admitted to Rose-Hulman and not to a particular degree program. They are free to change majors while at Rose-Hulman if their interests change. All they need do is notify the Registrar's Office of their new major, speak with the department head of the new major, and arrange for transfer of their academic advising folder from their advisor in the old major to their advisor in the new major. For students changing majors into computer science, the computer science and software engineering department head counsels the student as to how the courses the student has already taken will be applied to the graduation requirements in computer science. Typically, technical courses in the old major will count as free electives in computer science. However, electrical and computer engineering courses can usually be counted as technical electives. The department head will also assign the appropriate academic advisor for the student, based on the student's year in school and standing in the computer science major.

### Transfer Students

RHIT has established a policy for the admission of transfer students to the Institute. Our department follows these policies. In order to be admitted as a transfer student to RHIT, a student must be in good academic standing at his/her current college or university.

#### *Transfer of Courses from Other Schools*

RHIT establishes policies for transfer credit at the Institute level. Our department follows these policies. A transfer student may be given credit for equivalent courses taken elsewhere if a grade of "C" or better was earned, and the course is at the 100-level or above. The head of the department in which the courses would have been taught evaluates both courses and credits. This is done on a case-by-case basis. Transfer of engineering courses taken elsewhere can only occur if the student was enrolled in a program accredited by ABET; transfer of other courses (such as humanities and social sciences) can only occur it the students was enrolled in a college or university that is accredited by a nationally recognized accrediting agency, such as the North Central Association.

Usually the granting of transfer credit is a smooth process, since students often transfer in from ranked and recognized colleges and universities. If the school is not well known, however, transfer credits will only be approved after the appropriate department chairman has talked to the student about the course(s) being transferred, looked over the catalog description(s), and determined what text was used in the course(s) and how much of the book was covered. Final credit transfers are not made until it is determined that the student can progress satisfactorily using this prior background information.

Final acceptance of the credit is at the discretion of the engineering department head. **You may include here some department-specific tracking you do for these transfer students.

#### *Summer Courses Transferred to RHIT*

Students planning on taking courses during the summer at another institution and transferring them back to Rose-Hulman must secure prior approval from the department that the proposed course is acceptable and meets all requirements. The monitoring of this program is the responsibility of the academic advisor and the appropriate department chairman. The overall

monitoring of the transfer credit program is the responsibility of the department chairman and the Registrar. The Registrar assures performance and consistency among all academic departments.

**Admission with Conditions**

Rose-Hulman requires incoming students to have completed a year each of high school physics and chemistry. Students who have not satisfied these requirements may be admitted on the condition that they provide proof that they have completed the course(s) before they enroll at Rose-Hulman. The most common way that students satisfy the condition(s) is by taking the course(s) at a local community college during the summer.

**Advanced Standing**

Students who earn a grade of 4 or 5 on either of the computer science AP exams qualify for enrollment in CSSE 221, Fundamentals of Software Development Honors. Upon successful completion of this course, students will earn 4 quarter hours of credit for the course, plus an additional four quarter hours of credit for CSSE 120, Fundamentals of Software Development I. In addition, students who earned a 4 or 5 on the AB level of the computer science AP exam will earn four more quarter hours of credit for CSSE 230, Fundamentals of Software Development III.

**Admission into "Upper Division"**

No special admission requirements beyond what is required for admission to the Institute as a transfer student.

# Appendix VII.  Program Assessment Data

# TWiki . CSSE . CSSECourseCoordinationTeams

| Course | Terms | Course Coordinator | Reviewer 1 | Reviewer 2 |
|---|---|---|---|---|
| CSSE 120 Fundamentals of Software Development I | FWS | Mutchler | Merkle | Ardis |
| CSSE 220 Fundamentals of Software Development II | FWS | Wollowski | Merkle | Ardis |
| CSSE 221 Fundamentals of Software Development Honors | F | Mutchler | Merkle | Ardis |
| CSSE 230 Fundamentals of Software Development III | WS | Wollowski | Merkle | Ardis |
| CSSE 232 Computer Architecture I | FW | Chidanandan | Bagert | Chenoweth |
| CSSE 304 Programming Language Concepts | S | Wollowski | Bagert | Mutchler |
| CSSE 325 Fractals and Chaotic Dynamical Systems | S | Laxer | Bagert | Wollowski |
| CSSE 332 Operating Systems | WS | Chidanandan | Bagert | Chenoweth |
| CSSE 333 Database Systems | W | Clifton | Bagert | Chidanandan |
| CSSE 351 Computer Graphics | F | Laxer | Bagert | Mutchler |
| CSSE 371 Software Requirements and Specification | F | Bagert | Merkle | Laxer |
| CSSE 372 Software Project Management | W | Bagert | Merkle | Kaczmarczyk |
| CSSE 373 Formal Methods in Specification and Design | F | Ardis | Merkle | Laxer |
| CSSE 374 Software Architecture and Design I | W | Chenoweth | Merkle | Kaczmarczyk |
| CSSE 375 Software Construction and Evolution | F | Bagert | Merkle | Laxer |
| CSSE 376 Software Quality Assurance | S | Ardis | Merkle | Kaczmarczyk |
| CSSE 377 Software Architecture and Design II | S | Chenoweth | Merkle | Laxer |
| CSSE 404 Compiler Construction | S | Clifton | Bagert | Chidanandan |
| CSSE 413 Artificial Intelligence | F?? | Chenoweth | Bagert | Clifton |
| CSSE 432 Computer Networks | S | Ardis | Bagert | Boutell |
| CSSE 433 Advanced Database Systems | ?? | TBD | Bagert | Chidanandan |
| CSSE 442 Computer Security | S | Merkle | Bagert | Boutell |
| CSSE 451 Advanced Computer Graphics | W | Laxer | Bagert | Mutchler |
| CSSE 453 Topics in Artificial Intelligence | ?? | Chenoweth | Merkle | Clifton |
| CSSE 461 Computer Vision | ?? | Boutell | Merkle | Wollowski |
| CSSE 473 Design and Analysis of Algorithms | F | Merkle | Bagert | Chenoweth |
| CSSE 474 Theory of Computation | W | Merkle | Bagert | Chenoweth |
| CSSE 479 Cryptography | ?? | TBD | Merkle | Wollowski |
| CSSE 481 Web-Based Information Systems | F | Wollowski | Bagert | Mutchler |
| CSSE 494 Senior Thesis | FWS | | | |
| CSSE 497 Senior Project | FWS | Ardis/Chenoweth | Merkle | |

-- CurtClifton - 01 Mar 2006 - set term for 404

-- LarryMerkle - 20 Sep 2005 - Added coordinator for 221 (Mutchler)

-- LarryMerkle - 14 Sep 2005 - Removed 490 (not relevant this term)

-- LarryMerkle - 12 Sep 2005 - Added coordinators to replace Anderson and Mellor, added reviewers

-- DonBagert - 12 Sep 2005 - Changed coordinator for 371 to Bagert and 497 to Ardis/Chenoweth

-- CurtClifton - 12 Sep 2005 - Changed coordinator for 333 and 404

-- CaryLaxer - 12 Sep 2005 - Updated information for 325, 351, and 451

-- MarkArdis - 12 Sep 2005 - Changed coordinator for 497 to Ardis/Bagert/Chenoweth

-- MarkArdis - 12 Sep 2005 - Changed coordinator for 377 to Chenoweth

-- MarkArdis - 12 Sep 2005 - Changed coordinator for 432 to Ardis, specified term.

-- MarkArdis - 12 Sep 2005 - Specified terms for 37x courses.

-- MarkArdis - 12 Sep 2005 - Added names to 37x courses.

-- ArchanaChidanandan - 12 Sep 2005 - Changed CSSE 332 Course Coordinator from Mellor to Chidanandan

-- LarryMerkle - 12 Sep 2005 - Created page

- - - - - Revision r1.12 - 01 Mar 2006 - 16:12 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE120FundamentalsofSoftwareDevelopmentIFall200506**

# CSSE 120 - Fundamentals of Software Development I

# Course Assessment Plan

## November 30, 2005

## Instructor

Matt Boutell, Curt Clifton, and David Mutchler

## Recommendations from Previous Offering and Corresponding Changes

None

## Other Significant Changes from Previous Offering

Changed the IDE to Eclipse, updated version control system to Subversion, used Java 5.0, introduced unit testing, continued to increase the emphasis on UML class diagrams and design-first, did peer evaluation of the teaming experience more often and earlier than ever before, used more comprehensive sets of interviews for evaluation.

## Course description

This course introduces students to current practices of producing object-oriented software. Using a disciplined design process, students explore multi-threaded event-driven programming, the development of graphical user interfaces, and interaction among objects. Students design and implement software individually, in small groups, and in a challenging multi-week team project. This course presumes no prior programming experience.

## Learning objectives

**Students who successfully complete this course should be able to:**

1. Analyze and explain the behavior of simple programs involving:
    a. Fundamental non-object-oriented programming constructs, including:
        - Basic syntax and semantics of a higher-level language
        - Variables, types, expressions, and assignment
        - Simple I/O
        - Conditional and iterative control structures
        - Functions and parameter passing
        - Arrays
    b. Fundamental object-oriented programming constructs, including:
        - Basic syntax and semantics of a class
        - Fields, constructors and methods
        - The distinction between primitive types/variables and non-primitive types/variables
        - The relationship between the static structure of the class and the dynamic structure of the instances of the class (e.g. via Java's *new* operator)
        - Visibility modifiers (e.g. Java's *public*, *protected* and *private*) and mutability modifiers (e.g. Java's *final*)
        - The distinction between a static method/field and a non-static method/field
        - Implementing an interface
        - Extending a class

        - Responding to an exception

2. Design, implement, debug and informally test simple programs in an object-oriented programming language, including programs that:
    - Use the above fundamental programming constructs
    - Support a graphical user interface developed by using a GUI toolkit
    - Respond to user events
    - Use threads
    - Include "is-a" relationships among objects using a class hierarchy and inheritance
    - Use large-scale API packages

3. Apply decomposition techniques, including:
    - Using structured (functional) decomposition (breaking a function/program into subfunctions)
    - Encapsulating behaviors and data into classes
    - Using inheritance

4. Explain techniques that scale up to larger software development projects (and explain why they scale up), including:
    - Pair programming
    - Structured documentation (e.g., javadoc comments)
    - Implementing by using stubs
    - Implementing by using an iterative enhancement plan
    - Using code conventions (e.g., Sun's code conventions)
    - Decomposition techniques listed above
    - Using an integrated development environment
    - Using UML class diagrams
    - Unit testing using a tool like JUnit
    - Using simple metrics to evaluate quality of code
    - Using simple team building and management techniques like brainstorming, roles and task lists

5. Work effectively as members of a small team

## Description of Assessment Tools

- Project reports
    - Each of the major projects (e.g. BallWorlds?) has an online report which students complete as part of the project.
    - The reports have various questions which collectively assess various course objectives.

- Project and learning objective interviews
    - During the term, the instructor meets individually with each student.
    - The student discusses learning objectives and perhaps shows some of her project work to the instructor, under the instructor's direction.
    - This interview assesses various course objectives at a relatively shallow level.

- Capstone project interviews
    - During the capstone project, the instructor meets informally with each of the student teams.
    - After the capstone project, the instructor meets with each student individually.
    - During these interviews, the students demonstrate their understanding by referring to their work on their capstone project.
    - These interviews assesses various course objectives at a deeper level than the project interviews.

- Capstone project deliverables include:
    - From each team:
        - Problem statement, screen layouts and use cases that specify the project
        - An iterative enhancement plan by which to deliver the project
        - A task list that shows who is to do what, by when, and whether they did it
        - A UML class diagram for the project
        - Documented code produced in stages
    - From each student:

- An assessment of her team's effectiveness
- A reflection on what she learned from the capstone project
- For each major project deliverable, a brief peer review of it, e.g. a Likert-scale poll with these questions:
  - For each team member, including yourself, to what extent did that team member contribute to this deliverable?
  - For each team member, including yourself, to what extent do you think that team member understands the ideas evidenced by this deliverable?

## Course Assessment Matrix

1. Analyze and explain the behavior of simple programs
2. Design, implement, test, and debug simple programs in an object-oriented programming language
3. Apply decomposition techniques
4. Explain techniques that scale up to larger software development projects (and explain why they scale up)
5. Work effectively as members of a team

| **Assesment Tool** | **Objective** | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Project reports | X | | | | |
| Project interviews | X | | | X | |
| Capstone project interviews | | X | X | X | X |
| Capstone project deliverables | | X | X | | X |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

## Objective Evaluation

The course was fully successful according to the above criterion.

## Summary of Relevant Student Feedback

Students would like to see fewer slides used. While students appreciated an on-line text, they would prefer a text with more, and more realistic, examples and more actual code.

## Instructors' Subjective Evaluation

This course was extremely successful. Student performance on the capstone projects was the best ever.

## Recommendations for Next Offering

We made two major changes to the environment: Eclipse as the IDE and Subversion (supported by Tortoise Subversion and Subclipse) for version control. Both were major successes. We suggest their continued use and that new assistants be reminded to become comfortable with these tools very quickly. Eclipse was a wonderul tool -- we suggest that you take full advantage of its ability to do things like write the method stub for the method whose call you are typing. The use of Subversion with its two front-ends (Tortoise and Subclipse) was particularly delicate -- for now, ask one of us or Mike for help in setting this up. Eventually the process should be more automated (it is already partially automated).

The increased emphasis on UML class diagrams was very successful. It was at least part of the reason for the superb Capstone projects. (The other major reason was the continued emphasis on following the PROCESS of software development that was presented in class.) We suggest that you continue to emphasize the importance of UML class diagrams. Violet was a fine tool for drawing the class diagrams but meant that once students began to code, they were unwilling to go back and change the class diagram in Violet. We suggest that you look for a UML class diagram tool that plugs into Eclipse, does round-trip engineering, is reliable, and is simple to use. The latter two items are the problem -- we did not find tools that satisfy those items yet.

We experimented with using on-line self-assessment exercises for laboratory work. For these exercises, a student would answer a question and would then be presented with our suggested answer. Based on the suggested answer the student would assess their own understanding of the question. Students really liked this format and we think the immediate feedback was very helpful for their learning. We would recommend that future offerings use this style of self-assessment exercise for readings and class summary exercises also.

Some students disliked having a textbook (Stein) with an abstract presentation. We experimented with using Big Java (Horstmann), a more example-oriented book, as an optional text. We weren't very consistent with recommending specific readings from Big Java. The textbook issue should continue to be examined.

-- DavidMutchler - 1 Dec 2005 Added mention of Eclipse, added emphasis on UML

-- CurtClifton - 29 Nov 2005 Minor tweaks

-- MatthewBoutell - 29 Nov 2005

- - - - - Revision r1.4 - 01 Dec 2005 - 19:51 GMT - MatthewBoutell

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

# CSSE 220 - Fundamentals of Software Development II
# Rose-Hulman Institute of Technology
# Computer Science and Software Engineering Department

## Course Assessment Report

**Instructor**

Michael Wollowski

**Recommendations from Previous Offering and Corresponding Changes**

1. Add additional written exercises for objectives 5.
2. Add programming assignments and exam questions for generics.
3. Add additional programming assignments on functors
4. Give a programming assignment on databases, or consider having a database component for the final project.
5. Keep more detailed records of the items listed in objective 3.
6. Fine tune the hardness of the written exercises and the exam questions for objective 4. This should be done in light of the fact that CSSE 230 will cover an increased amount of analysis.

I implemented recommendation number 2, 4, 5 and 6. There was not sufficient classtime to implement 1 and 3.

**Other Significant Changes from Previous Offering**

Reviewed a good amount of CSSE 120 materials.

**Learning Outcomes**

Students who successfully complete this course should be able to:

1. Develop software that incorporates the following techniques:
    - Inheritance
    - Interfaces
    - Polymorphism
    - Casting
    - Function objects
    - Generics
    - Collections
2. Develop software in each of the following application areas:
    - Software that interacts with a database
    - Software that communictes over a network
    - Software that uses a graphical user interface
3. Perform the following steps of the software development cycle effectively:
    - Requirements elicitation
    - Design expressed as UML class diagrams
    - Implementation using an iterative enhancement plan and stubs/documentation before coding
    - Unit, system, and acceptance testing

4. Analyze the asymptotic worst, best, and average case run times of simple algorithms.
5. Select basic data structures (e.g, arrays, sequential lists, linked lists, stacks, queues, hash tables and trees) based on the time and space complexity of typical operations.
6. Develop software that uses recursive methods and data structures.

**Description of Assessment Tools**

**Written exercises:** Short thought problems, mathematical analyses, or algorithm-design exercises.

**Programming assignments:** Several programming assignments of varying size and length, including a mini-project.

**Projects:** One large team project.

**Exams:** Two midterm exams and one final exam.

**Course Assessment Matrix**

| | Outcomes | | | | | |
|---|---|---|---|---|---|---|
| Tool | 1 | 2 | 3 | 4 | 5 | 6 |
| Written exercises | | | | X | X | |
| Programming assignments | X | X | X | | | X |
| Projects | | X | X | | | |
| Exams | X | | | X | X | X |

**Success Criteria**

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

**Objective Evaluation**

The success criterion holds for all tool-outcome pairs except for the following:

- Written exercises: Objectives 4 and 5. They were not assessed. As indicated below, we spent a good amount of time on review which took up time that had to be sacrificed elsewhere. Please note that students did fine on these two objectives during the second exam and the final exam. There were three rather elaborate questions on the exams, totaling 63 points out of 200 (combined.)
- Programming assignments: Objectives 1, 2(a), and 6. Several students did not turn in those assignments and received 0 credit for them. The averages among the students who did turn in those assignments were in the 90% range.

**Summary of Relevant Student Feedback**

- Some students disliked the material covered, one of them even changing majors.
- Some students felt that I should have enforced deadlines more rigorously

**Instructors' Subjective Evaluation**

This is the second time I taught this course where students felt that they were not adequately prepared for this course. Based on the discussions I had with my students and based on my observations of their programming skills and conceptual knowledge of the CSSE 120 materials, I agree with them. In general, students felt that they did not learn enough in CSSE 120 and would like to learn more.

In light of the above realization, I had to improvise quite a bit, reviewing materials that students should have known coming into this course, and extending deadlines for assignments so as to give students additional time to learn materials in which they were deficient.

In the defense of this offering, a good number of the students in this class did not apply themselves, resulting in 4 students out of 13 receiving penalty grades. Two of those students since withdrew or were withdrawn from Rose-Hulman.

**Recommendations for Next Offering**

- Reduce the amount of materials covered or reduce the depth of coverage.
- Consider introducing or reviewing a good amount of the material to be covered in CSSE 120.

-- MichaelWollowski - 02 Jun 2006

----- Revision r1.2 - 07 Jun 2006 - 14:41 GMT - MichaelWollowski

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE351ComputerGraphicsFall200506**

# CSSE 351 Computer Graphics

# Course Assessment Report

# December 1, 2005

## Instructor

Cary Laxer

## Recommendations from Previous Offering and Corresponding Changes

- Students wanted more quizzes and homeworks. I added an additional quiz this year, but no additional homework.

## Other Significant Changes from Previous Offering

- None

## Learning Objectives

Students who successfully complete this course should be able to:

1. Develop interactive computer graphics applications using a modern graphics API
2. Model geometric objects, curves, and surfaces
3. Project a three-dimensional scene onto a two-dimensional viewing surface
4. Apply lighting and shading to a scene
5. Describe the mathematical foundations of computer graphics (e.g. Bresenham's algorithm, transformation matrices, projection matrices)
6. Apply a texture map to an object

## Description of Assessment Tools

### Problem Sets

Homework assignments, each containing a few problems on computer graphics concepts that students are to answer on paper, without writing any software.

### Quizzes

Short, in-class examinations allowing students to show individual mastery of key computer graphics concepts.

### Projects

Several one- to two-week projects that students write either individually or in teams of 2-3 students using OpenGL? and C or C++. Each successive project requires use of additional computer graphics concepts that have been learned in class.

### Final project

One three-week project that students do in a team of 3-4 students that is open ended, allowing students to showcase their computer graphics skills and creativity.

### Course Assessment Matrix

| | **Objective** | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| **Problem sets** | | | | | X | |
| **Quizzes** | | | | | X | X |
| **Projects** | X | X | X | X | | |
| **Final project** | X | X | X | X | | X |

### Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn passing grades in the course, the minimum grade on the portions of the assessment tools that are relevant to the learning objective is in the passing range.

## Objective Evaluation

- Objective 5 (mathematical foundations) was not successful for two students (out of 34) on problem sets and for half the class on the quizzes.
- Objective 6 (texture mapping) was not tested on a quiz.

## Summary of Relevant Student Feedback

- Students enjoyed the course and the projects. Some would like more homework exercises. A couple of students felt there was too much material to cover in the course.

## Instructors' Subjective Evaluation

- The course was, in general, successful, with every student earning a passing grade. Perhaps one of the

quizzes was too long for the time allotted, which resulted in the poor performance.

# Recommendations for Next Offering

- Try to do a couple more homework sets covering the mathematical algorithms underlying computer graphics.

-- CaryLaxer - 01 Dec 2005 - completed the report.

-- CaryLaxer - 28 Nov 2005 - started the report.

- - - - - Revision r1.2 - 01 Dec 2005 - 16:25 GMT - CaryLaxer

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

# CSSE 371 Software Requirements and Specification

# Course Assessment Report

# May 15, 2006

## Instructor(s)

Don Bagert (two sections)

## Significant Changes from Previous Offering

- Instead of each project team working on a "play" project with the course instructors as clients, this time each team worked on an actual project (i.e. one intended for actual use) with a Rose-Hulman faculty or staff member as a client. These projects will be finished in CSSE 372 (Software Project Management).
- Instead of “paper” interface prototypes, the teams developed actual user interface prototypes that users of the system being developed can evaluate and provide feedback.

## Learning Outcomes

Students should be able to:

1. Explain the role of requirements engineering and its process
2. Use analysis techniques to develop a problem statement
3. Use multiple techniques to elicit requirements from stakeholders
4. Take the elicited requirements and develop a specification with functional and non-functional requirements
5. Negotiate with the client and other stakeholders regarding priorities and scope
6. Use quality assurance techniques to verify that requirements are: verifiable, traceable, measurable, testable, accurate, unambiguous, consistent, and complete
7. Know how to manage requirements
8. Develop and use user interface prototypes to validate requirements

## Description of Assessment Tools

Project – Each student is part of a team project with a Rose-Hulman faculty or staff member as a client; each team has a different project.

Homeworks – Six homework assignments performed a various times throughout the term are used to apply and reinforce material from the course lectures and discussions.

Examinations – Three exams are used to test the student’s knowledge and capabilities regarding the course material.

## Course Assessment Matrix

| **Assessment Tool** | **outcome** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| Projects | X | X | X | X | X | | | X |
| Homeworks | | | X | X | X | X | X | X |
| Examinations | X | X | | | | X | X | |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

## Objective Evaluation

The success criterion holds for all tool-outcome pairs.

## Summary of Relevant Student Feedback

- The students liked the "real" projects with the RHIT clients; however, the fact that they did not see a finished project hurt their perception of the value of the requirements specification and user prototype development.
- The students felt that the user interace portion of the course at the end was rushed.
- The students wanted more lecture and less discussion time.
- The students would have liked a couple of guest lecturers from industry.

## Instructors' Subjective Evaluation

- I am mostly satisfied with the students' learning and experience in the course. I do believe that in retrospect the students will better see the course's value after CSSE 372 and internship/job experience.

## Recommendations for Next Offering

- Allocate more time for lectures in general and for user interface prototyping in particular.
- Bring in a couple of guest lecturers from industry.

-- DonBagert - 15 May 2006

- - - - - Revision r1.2 - 31 May 2006 - 14:34 GMT - DonBagert

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE373FormalMethodsInSpecificationAndDesignFall200506**

# CSSE 373 Formal Methods in Specification and Design

# Course Assessment Report

# DRAFT

## Instructor

Mark Ardis

## Recommendations from Previous Offering and Corresponding Changes

- Change project to get more practice applying methods.

## Other Significant Changes from Previous Offering

- (none)

## Learning Objectives

Students who complete this course should be able to:

1. demonstrate formal correctness of simple procedures
2. construct formal models of sequential software systems
3. construct formal models of concurrent software systems
4. verify attributes of formal models
5. comprehend the costs and benefits of formal methods.

## Description of Assessment Tools

Daily Quizzes - short (5-10 questions) quizzes completed during class

Homework Assignments – exercises in the application of formal methods

Project - team project in specification and analysis of a safety-critical application

Exams – three one-hour exams.

## Course Assessment Matrix

| | **Objective** | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Quizzes** | | | | | X |
| **Homework** | X | X | X | X | |
| **Project** | | X | X | | |
| **Exams** | X | | | X | |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant

to the learning objective is in the proficient range.

## Objective Evaluation

The success criterion holds for all tool-objective pairs.

## Summary of Relevant Student Feedback

- Students like the 3-stage approach: quizzes to learn new topics, homework to expand on that learning, project to master technique and tie topics together.

## Instructors' Subjective Evaluation

- I am satisfied with the students' learning and experience in the course.

## Recommendations for Next Offering

- Some students struggled a bit with the program proofs. A few more examples would help there, or a special help session might be scheduled.
- The Z type checker ZTC is hard to learn. It would be a good idea to spend a little more time in class working through a few simple examples. If this were scheduled as an in-class lab it would force students to confront this earlier.
- Some students found the telephone example confusing. That was the point: the original specifications are imprecise, and formal methods solve that. This point should be made when assigning those exercises next time.

-- MarkArdis - 02 Feb 2006

- - - - - Revision r1.1 - 02 Feb 2006 - 15:45 GMT - MarkArdis

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE413ArtificialIntelligenceFall200506**

# CSSE 413, Artificial Intelligence

# Course Assessment Report

# DRAFT – November 9, 2005

## Instructor(s)

Steve Chenoweth (two sections)

## Recommendations from Previous Offering (Fall 2004-5) and Corresponding Changes

- Provide more explicit algorithms for major assignments. (The book, Russell & Norvig, generally only gives high level algorithms, no coding examples.) -> We added actual coding examples related to most of the lab assignments.
- Give additional assistance to students first learning Scheme and Lisp. (Typically 80% have had experience with one of these languages before, mostly from CSSE 304.) -> We emphasized help time available, and most of those needing it did make use of this offer.
- Maintain detailed feedback on lab assignments. (This has been a major feature of the course the past 2 years - for the instructor to do extensive testing on the multiple projects each student turns in. And these projects generally are allowed to vary with student interest.) -> We repeated this, with emphasis on instructor feedback in areas where students did some unique work (something that takes a lot of instructor time to analyze thoroughly).
- Maintain coverage of a broad range of AI problem-solving methods (from student feedback). -> Once again, major labs were run on a range of methods, and the final one was left entirely to student choice.
- Note: This is an elective course whose description is very general. It is taught with objectives that vary at instructor discretion.

## Learning Objectives

Students who complete this course in the 2005-06 school year will be able to:

1. Describe the fundamental design alternatives for AI and for intelligent agents.
2. Use common tools for building AI systems.
3. Develop a system to do heuristic search in a significant domain such as the Traveling Salesperson Problem.
4. Build a knowledge representation model and use it for decision making.
5. Create an AI program for which quality results and timeliness are both critical.
6. Apply the concepts of learning and probability to AI.
7. Debate the philosophical and ethical issues surrounding AI.

# Description of Assessment Tools

Project – Four two-week individual or team projects (done primarily outside scheduled class time). In these projects the students develop systems to test their understanding of the concepts described in the objectives. One of the projects is competitive, and on one the grade includes peer evaluation.

Homeworks – Individual assignments, ranging from short essays on current AI topics to assignments to learn AI programming languages. Debate results (see objective 7) are graded and count as a homework.

Biweekly quizzes – Five approximately 30 minute short essay quizzes. The first three are closed book, done in class. The final two are open book, take home.

Daily quizzes – Twenty five 5-10 minute short answer quizzes, done each day new material is discussed in class.

# Course Assessment Matrix

| | **Objective** | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Projects** | | X | X | X | X | | |
| **Homeworks** | | X | | | | | X |
| **Biweekly Quizzes** | X | | | | X | X | |
| **Daily Quizzes** | X | | X | X | | X | X |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

# Objective Evaluation

All objectives were met by all students successfully completing the course. One student failed and others dropped the course before the drop date, primarily when it became clear they could not complete the projects (and so meet these objectives).

# Summary of Relevant Subjective Student Feedback (volunteered on biweekly quizzes)

- Over 2/3 of students reported the pace was good. (This was covering 17 of the 27 Chapters in Russell & Norvig, with lab assignments involving at least 10 of those chapters -- more if the students' final project involved learning.)
- "Things are getting even more interesting and useful, so the class is becoming more enjoyable."
- "Homeworks going good, I think. I’m having a good time on them."
- "Ok, but busy classes add difficulty in keeping up."

- "So much reading – at least it's interesting." (Most such comments related to the volume of reading, or to a particular lab.)
- " The game was stressful because I understood the algorithm well but my program would get tied up on little things – doing a program now that links to my M.E. stuff is very cool, as are G.A.'s." (From an ME major.)

# Instructor's Objective and Subjective Evaluation

- Additional examples of coding were provided on some of the labs -- in Scheme and in Java. These were reported as very useful, even though they were (obviously) not specifically solving the same problem as the lab assignment.
- Help sessions by the prof and by the lab assistant were heavily advertised. Those who came to these reported that they were very helpful.

# Recommendations for Next Offering

- The major labs in the course take some students much longer to complete than others. (E.g., 4 - 6 hours median, but 10 - 40 hours for some outliers, even with help.) Since the level of difficulty is rated as about right by the majority, there is an interesting dilemma about how to handle this for those struggling with the labs. Lab difficulty also is a stated cause of course dropouts.
- Since teamwork is not an explicit objective of the course, we experimented with allowing individuals or teams to work on the final two projects. (The expectations were adjusted to make this equitable.) This appeared to work well, and make the course especially interesting to students who had unique interests, etc. We believe this would be worth trying again.
- IRPA is doing a study of these Fall 2005-6 classes to correlate factors such as outside interests and learning preferences with success in this class. When available, those results should provide additional recommendations. Ideally, a continuation of this study can be a part of next year's AI classes, regardless of who the instructor is.
- Other instructors may choose to go with a less broad scope. This approach will be interesting, in retrospect, to compare with the broad approach we've used the past three years. The major issue may be that many students come into the class having a strong interest in some particular niche area of AI, such as game playing, Genetic Algorithms or Neural Networks, so ignoring those will be a disappointment to them. We have done further accommodation on this point by making the final project one that is entirely at the students' discretion.
- The objective to "Use common tools for building AI systems" primarily means common development environments. In the courses over the past three years, this has meant projects in Scheme, Java and Lisp, with an introductory discussion of Prolog. The rationale is that students going on to study AI in graduate school, of which we've had several, are likely to use an "AI language" like Scheme or Lisp, whereas students going to work in software development will almost surely have to create any AI features in an industrial language like Java. The choices here are likely to vary according to the perspective of who teaches the class, among other variables. (At Berkeley, the course would have to be in Lisp. At MIT, it would have to be in Scheme. At Iowa State, Dr. Clifton tells me, it would be in Java. So far, we are leaving language as a free variable for this elective course.)

-- SteveChenoweth - 25 Jan 2006 -- SteveChenoweth - 09 Nov 2005

- - - - - Revision r1.3 - 26 Jan 2006 - 01:40 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

# CourseAssessmentReportCSSE481WebBasedInformationSystemsFall200506

# CSSE 481 - Web-based Information Systems
# Rose-Hulman Institute of Technology
# Computer Science and Software Engineering Department
# Course Assessment Report

**Instructor**

Michael Wollowski

**Recommendations from Previous Offering and Corresponding Changes**

None, first time assessed.

**Other Significant Changes from Previous Offering**

None, first time assessed.

**Course Outcomes**

Students who successfully complete this course, should be able to:

1. Stay current in the area of the world-wide web.
2. Identify and propose applications which advance the web.
3. Develop applications and perform research which advances the web.
4. Present their own work in a scientific manner.

**Description of assessment tools**

**Class participation.** Class participation. This includes participation in class discussion, presentations of papers, project proposal, and interesting web-sites.

**Project proposal.** Write-up of project proposal following typical format and contents. Peer review of a project proposal, following typical fomat of peer review.

**Project presentation.** Oral class presentation of project work, following standard conference length and format.

**Project report.** Formal paper of project work in the format and length of a conference paper.

**Homework.** These are typically reading summaries and programming assignments.

**Course Assessment Matrix**

| Tool | Objective | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| Class participation | X | X | | |
| Project proposal | | X | | |

| | | | | |
|---|---|---|---|---|
| Project presentation | | | X | X |
| Project report | | | X | X |
| Homework | X | | | |

**Success Criteria**

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

**Objective Evaluation**

The success criterion holds for all tool-outcome pairs.

**Summary of Relevant Student Feedback**

- Students were unhappy with the discussions we have had in this course, one student going as far as suggesting that this was a Humantities course.
    - Comment from David Mutchler: this is a ***positive*** recommendation for the course, not a negative comment!
- Students liked the project a lot and it helped their learning.
- Students would like to have more technical materials covered in class.
- Students would like to have more structure to this class.

**Instructors' Subjective Evaluation**

Students did not react well to the open-ended nature of this class. I was hoping to inspire creativity and give students time to think about where this field may be headed by having a lot of discussions of papers which address current interest and by asking students to think creatively about the future. This did not work well.

**Recommendations for Next Offering**

- Make this a considerably more technical course which focuses on technologies more than on future directions and creative thought.
- It is fine to keep some discussions about where the field may be headed, but for now, focus on technical skills.
    - Comment from David Mutchler: I support whatever the instructor thinks as to how technical or non-technical this course should be. However, I would encourage the instructor not to overreact to students who comment that "this is a humanities course" or "this course should have more structure". It is important to remember that we must ***stretch*** our students. Yes, many of our students are uncomfortable with a course that demands that they think creatively within an open-ended structure. Yes, many of our students resist courses like this that are interdisciplinary (here, a crossover with humanities). But that just makes it ***more*** important that we offer more such opportunities for them and encourage them to take advantage of those opportunities.

-- MichaelWollowski - 05 Jun 2006

- - - - - Revision r1.2 - 05 Jun 2006 - 16:32 GMT - DavidMutchler

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE120FundamentalsofSoftwareDevelopmentIWi**

# CSSE 120 Fundamentals of Software Development I

# Course Assessment Report

## PROPOSED - May 17, 2006

## Instructor

Lisa Kaczmarczyk (2 sections)

## Recommendations from Previous Offering and Corresponding Changes

(summarized from text into bullet points)

- Continue using Eclipse. (Done)
- Continue using Subversion. (Not Done)
- Remind course assistants to become comfortable with Eclipse/Subversion as early as possible. (Done for Eclipse)
- Continue to emphasize the importance of UML diagrams. (Done)
- Look for a UML class diagram tool that plugs into Eclipse, does round-trip engineering, is reliable, and is simple to use. (Not Done)
- Continue using on-line self-assessment exercises. (Done)
- Continue to examine the use of the Big Java textbook by Horstmann. (Done)

## Other Significant Changes from Previous Offering

None

## Learning Outcomes

Students who successfully complete this course should be able to:

**Students who successfully complete this course should be able to:**

1. Analyze and explain the behavior of simple programs involving:
    a. Fundamental non-object-oriented programming constructs, including:
        - Basic syntax and semantics of a higher-level language
        - Variables, types,expressions, and assignment
        - Simple I/O
        - Conditional and iterative control structures
        - Functions and parameter passing
        - Arrays
    b. Fundamental object-oriented programming constructs, including:
        - Basic syntax and semantics of a class
        - Fields,constructors and methods
        - The distinction between primitive types/variables and non-primitive types/variables
        - The relationship between the static structure of the class and the dynamic structure of the instances of

the class (e.g. via Java's *new* operator)
    - Visibility modifiers (e.g. Java's *public*, *protected* and *private*)and mutability modifiers (e.g. Java's *final*)
    - The distinction between a static method/field and a non-static method/field
    - Implementing an interface
    - Extending a class
    - Responding to an exception

2. Design, implement, debug and informally test simple programs in an object-oriented programming language, including programs that:
    - Use the above fundamental programming constructs
    - Support a graphical user interface developed by using a GUI toolkit
    - Respond to user events
    - Use threads
    - Include "is-a" relationships among objects using a class hierarchy and inheritance
    - Use large-scale API packages

3. Apply decomposition techniques, including:
    - Using structured (functional) decomposition (breaking a function/program into subfunctions)
    - Encapsulating behaviors and data into classes
    - Using inheritance

4. Explain techniques that scale up to larger software development projects (and explain why they scale up), including:
    - Pair programming
    - Structured documentation comments
    - Implementing by using stubs
    - Implementing by using an iterative enhancement plan
    - Using code conventions (e.g., Sun's code conventions)
    - Decomposition techniques listed above
    - Using an integrated development environment
    - Using UML class diagrams
    - Unit testing using a tool like JUnit
    - Using simple metrics to evaluate quality of code
    - Using simple team building and management techniques like brainstorming, roles and task lists

5. Work effectively as members of a small team

# Description of Assessment Tools

- Each of the major projects (e.g. Ball Worlds) has an online report which students complete as part of the project.
- Several projects and text reading assignments have written reports which students hand in individually.
- The reports have questions which collectively assess various course objectives.
- Mini project deliverables include:
    - From each team:
        - Problem statement, screen layouts and use cases that specify the project
        - An iterative enhancement plan of how to implement the project
        - A schedule by which to deliver the project
        - A task list that shows who is to do what, by when, and whether they did it
        - A UML class diagram for the project
        - Documented code produced in stages

        - A one page Executive Summary upon completion
    - From each student:
        - An assessment of her/his team's effectiveness
        - A reflection on what s/he learned from the Mini project

- Capstone project deliverables include:
    - From each team:
        - Problem statement, screen layouts and use cases that specify the project
        - An iterative enhancement plan of how to implement the project
        - A schedule by which to deliver the project
        - A task list that shows who is to do what, by when, and whether they did it
        - A UML class diagram for the project
        - Documented code produced in stages
        - A one page Executive Summary upon completion
        - A final presentation of their project, including a live demonstration
    - From each student:
        - An assessment of her/his team's effectiveness
        - A reflection on what s/he learned from the Capstone project

# Course Assessment Matrix

1. Analyze and explain the behavior of simple programs
2. Design, implement, test, and debug simple programs in an object-oriented programming language
3. Apply decomposition techniques
4. Explain techniques that scale up to larger software development projects (and explain why they scale up)
5. Work effectively as members of a team

| **Assesment Tool** | **Outcome** | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Project reports | X | X | | | |
| Mini project deliverables | | | | X | |
| Capstone project final presentations | X | | X | X | X |
| Capstone project deliverables | | X | X | | X |

# Success Criteria

The course will be considered fully successful if the following statement holds for every outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

# Objective Evaluation

See discussion under Instructor's Subjective Evaluation

# Summary of Relevant Student Feedback

- Many students felt overwhelmed by the material.
- Many students wanted more one on one assistance during class.
- Many students felt that the class asked them to perform too much problem solving on their own.
- Many students wanted to write their own simple program earlier in the class.
- Many students enjoyed the Capstone experience.

## Instructor's Subjective Evaluation

- The majority of the project pairs and project teams performed adequately on their designs and programs
- With signficant instructor assistance, most of the teams produced acceptable Project Reports
- With significant instructor assistance, most of the Mini and Capstone teams produced acceptable designs and implemention
- The pair and team setup of most tasks made it very difficult to verify that each student was able to successfully design, implement, debug and informally test code
- In-class observation and one on one conversations with students suggested that many students were unable to successfully design, implement, debug and informally test code on their own

- There were no CSSE majors in either section; it was difficult to convince them the class was relevent.
- The material as designed asks the students to take a lot on faith.
- There was a significant academic dishonesty situation in the middle of the term which negatively impacted one of the sections.
- There were quite a few students who skated by by using their classmates with greater technical ability.

## Recommendations for Next Offering

- The class needs more individual accountability.
- The class needs to impart a sense of accomplishment to the students early on - with coding.
- The students should get to write their own simple code early on to acheive this sense of accomplishment.
- The overall immersion approach to this class should be changed, to reduce the appearance of "busy work" but more importantly, to reduce the "overwhelmed" feeling, and improve learning.
- Use Big Java as the primary textbook and continue working to integrate appropriate readings into the course.
- Do not use the online Stein textbook.

-- LarryMerkle - 19 May 2006 - Formatting changes

-- LisaKaczmarczyk - 03 May 2006

- - - - - Revision r1.6 - 27 May 2006 - 13:30 GMT - LisaKaczmarczyk

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE220FundamentalsOfSoftwareDevelopmentIIWinter200506**

# CSSE 220, Fundamentals of Software Development II

# Course Assessment Report

## Instructor

David Mutchler (2 sections)

## Recommendations from Previous Offering and Corresponding Changes

- Reduce the amount of materials covered or reduce the depth of coverage -- perhaps done.
- Consider introducing or reviewing a good amount of the material to be covered in CSSE 120 -- done, although perhaps not adequately.

## Other Significant Changes from Previous Offering

- I used DyKnow and Tablet PCs as a pedagogical tool. This was my first extended experience with DyKnow.
- I took advantage of two opportunities to do something special this term -- an interdisciplinary collaboration.

[From an article presented at The First Workshop on the Impact of Pen-based Technology on Education (WIPTE 1), April 6-7, 2006, Purdue University]:

The Challenge X project (crosslisted as ME and ECE) is a multi-year, interdisciplinary, real-world learning experience in which Rose-Hulman is participating ...

“Visual/Rhetoric for a Mobile Age” (RH 339) ... focuses on teaching students to write for and about pen-based devices, including Tablet PCs and iPAQs. Most documentation projects result in paper products or electronic versions that are viewable on large screens. This class, in contrast, asks students to write documentation that is viewable on small iPAQ screens and relies on pen-based navigation.

... – a joint, interdisciplinary solution in which handheld, pen-based devices (HP iPAQs) play a central role, as follows: (The hard part was not coming up with the solution. Rather, the hard part was finding the fortuitous opportunity for three faculty members from three departments to talk about their very different interests and needs.) • The Challenge X team members each use a pen-based device (HP iPAQ) for project management. • Challenge X students are customers for the CSSE and RH students. • Early in the term, the students in both of the latter courses solicit the requirements for the pen-friendly custom software that the Challenge X students want for their project management on iPAQs. • CSSE students design and develop the software. In doing so, they encounter many of the same issues that they would encounter in developing software for other pen-based devices like Tablet PCs. • Meanwhile, RH students learn how to write for pen-based devices by writing the documentation for how to use that custom software that is viewable on the iPAQs. This includes documentation concerning the use of the pen in conjunction with the software. They relate to the CSSE students as clients and the Challenge X students as the end users, interacting with both groups. • Finally, all three groups are assessed in part on how well they understood the work of the other groups, thus helping to ensure that students recognize the interdisciplinary nature of this project.

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Develop software that incorporates the following techniques:
    - Inheritance
    - Interfaces
    - Polymorphism
    - Casting
    - Function objects
    - Generics
    - Collections
2. Develop software in each of the following application areas:
    - Software that interacts with a database

    - Software that communictes over a network
    - Software that uses a graphical user interface

3. Perform the following steps of the software development cycle effectively:
    - Requirements elicitation
    - Design expressed as UML class diagrams
    - Implementation using an iterative enhancement plan and stubs/documentation before coding
    - Unit, system, and acceptance testing
4. Analyze the asymptotic worst, best, and average case run times of simple algorithms.
5. Select basic data structures (e.g, arrays, sequential lists, linked lists, stacks, queues, hash tables and trees) based on the time and space complexity of typical operations.
6. Develop software that uses recursive methods and data structures.

## Description of Assessment Tools

- Software design, development and analysis exercises: Several programming assignments of varying size and length, sometimes accompanied by run-time analysis.
- Team project and individual interviews: One large team project, along with individual interviews at the end of the term, to discuss what the student learned from the programming assignments and the team project.

## Course Assessment Matrix

| | **Objective** | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| **Progamming assignments** | X | X | X | X | X | X |
| **Team project and individual interviews** | X | X | X | X | X | X |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcomes pair selected above:

> Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcomes is in the proficient range.

## Quantitative Evaluation

The success criteria hold for all tool-outcome pairs.

## Summary of Relevant Student Feedback

"The course built on the foundations of what was taught in the previous course and helped to further my understanding of concepts that I was exposed to in the previous course. The use of DyKnow in the course also helped with note-taking and to present the material."

"this course has interesting material"

"The joint class work was challenging and brought real life issues into that class (such as communication, confusion, etc.). Dyknow was also a useful tool in teaching concepts in class."

## Instructor's Subjective Evaluation

The unique project experience dominated this term's offering of CSSE 220. It was a great project and a great experience for the students. Of course, there were hitches (not enough interaction with the RH 339 students, and too "big" a project without an overall architecture that the instructor would provide). However, these hitches were more than made up for by the opportunity to participate in a real-world, interdisciplinary project with real clients.

Students appreciated my focus on learning, not grading. For almost all students that worked very well. For about 15% of the students, it worked very poorly, as the students were not sufficiently mature to work effectively without exams and the like as "sticks" to keep them working.

One section was at 8 a.m. -- attendance for several of the students was very poor. Even the 10 a.m. section had attendance problems on Wednesdays.

I did not use DyKnow as much as I would have liked to. However, I was VERY pleased with the uses that I did make of it. Students shared that enthusiasm -- in a post-course survey, 100% of the students felt that DyKnow/Tablets improved their learning experience.

## Recommendations for Next Offering

- Try the project again, but with an overall architecture provided.
- This course is a great course, with excellent projects that were mostly developed by Mike Wollowski. Keep doing them.
- If the instructor is interested in trying DyKnow, certain aspects of this course work very nicely with DyKnow.
- This course is very uneven in the quality of the students it gets from term to term. That makes it a hard course to teach. I agree with the previous CAR that we need more review/teaching of CSSE 120 topics -- while some students learn all the material of CSSE 120, not all CSSE 120 students learn all the material of CSSE 120. Since CSSE 220 builds so directly on CSSE 120, anything missed from CSSE 120 haunts the CSSE 220 student. One concrete way to address this challenge is: 1. Tie the "review project (Lights Out)" directly to CSSE 120 learning outcomes and do a test or interview after or during the Lights Out project to make sure that all CSSE 220 students really understand all the concepts covered by Lights Out (and don't merely "complete the project"). 2. Add another review project that covers the remaining learning outcomes from CSSE 120, again doing an exam or interview during or after that project to ensure that students truly master all the learning outcomes.

## Revision History

-- DavidMutchler - 05 Jun 2006

- - - - - Revision r1.2 - 08 Jun 2006 - 13:45 GMT - DavidMutchler

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE230FundamentalsofSoftwareDevelopmentIIIWinter200506**

# CSSE 230 Fundamentals of Software Devlopment III

# Course Assessment Report

# Winter 2005-2006

## Instructor

Matt Boutell (2 sections)

## Recommendations from Previous Offering and Corresponding Changes

- There was also no major term project, following departmental discussion of this course in 2004-2005.

## Other Significant Changes from Previous Offering

- I tried to make the class more interactive by replacing some segments of lecture with hands-on activities, ranging from implementation of algorithms we discussed in class to guided reading assignments.

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Describe the classical data structures (list, stack, queue, tree, priority queue, hash table, graph, set, dictionary) and understand the issues involved in implementation choices for each.
2. Analyze and prove the asymptotic worst, best, and average case run times of algorithms, including non-trivial recursive algorithms, using Big O, little o, Omega, and Theta notation.
3. Analyze and prove lower-bound results on problems like comparison-based sorting and understand the implications of lower bound results.
4. Understand and analyze classical sorting, graph and tree-balancing algorithms.

## Description of Assessment Tools

Exams – Two exams and a final exam.

Programming assignments. Five to seven short (typically less than 500 lines of code each) programming assignments, at least two of them to be done individually, some by groups of two or three students.

Written Homework – Eight to twelve take-home assignments on various topics covered in class. An assignment typically consists of four to six problems from the textbook on various aspects of algorithms, data structures, and analysis.

## Course Assessment Matrix

| | **Objective** | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| **Exams** | X | X | X | X |
| **Written Homework** | | X | X | |
| **Programming Assignments** | X | | | X |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

## Objective Evaluation

No data was collected, because I was too busy in my first time teaching this course to do so.

## Summary of Relevant Student Feedback

- While many students said they felt that the workload was too high, they liked the course, especially the assignments. They also liked the in-class activities, and asked for more. Students did not like the reading quizzes and thought they were too difficult, given that they had not yet had a chance to ask questions about the material in class. They also questioned the heavy use of proof and induction, and felt they were not adequately prepared to complete some of the homework problems.

## Instructors' Subjective Evaluation

- I am very pleased with how the course went in my first time teaching it. In my opinion, the students sucessfully learned the material in outcomes 1, 2, and 4. However, I did not assign enough material on lower-bounds to assess outcome 3. An assignment using data structures from the Collections interface should have been assigned early in the course. I thought the in-class activities reinforcing the theory given in lecture were successful.

## Recommendations for Next Offering

- Make the workload more manageable by coordinating the due dates of programming and written assignments such that they don't occur on the same day.
- Continue to look for ways to reinforce the theory using hands-on in-class activities.
- Add more justification for proof and induction.
- Provide more examples of strong induction.

-- MatthewBoutell - 12 May 2006 - Completed instructor's evaluation, and gave more thorough summary of student feedback.

-- MatthewBoutell - 09 May 2006 - Minor tweaks in wording, objective-->outcome.

-- MatthewBoutell - 08 Mar 2006 - Initial post.

- - - - - Revision r1.7 - 17 May 2006 - 13:46 GMT - MatthewBoutell

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE332OperatingSystemsWinter20052006**

# CSSE 332 Operating Systems

# Course Assessment Report

## Instructor(s)

Archana Chidanandan (two sections)

## Recommendations from Previous Offering and Corresponding Changes

- Introduction to C was strengthened with more in-class exercises, programming projects.
- Included a scheduling project
- Included process management in course objectives
- Team size smaller reduced to two or three.

## Other Significant Changes from Previous Offering

## Learning Objectives

Students who complete this course should be able to:

1. Explain the objectives and functions of modern operation systems.
2. Describe how operating systems have evolved over time from primitive batch systems to sophisticated multiuser systems.
3. Analyze the tradeoffs inherent in operating system design.
4. Explain the hardware support necessary to implement security in operating systems.
5. Analyze the performance of preemptive and non-preemptive scheduling algorithms.
6. Explain the need for process synchronization and demonstrate the proper use of common synchronization techniques.
7. Explain memory hierarchy, including virtual memory, caching, paging and segmentation, and analyze memory performance and cost-performance tradeoffs.
8. Explain the different approaches to file organization and their strengths and weaknesses.
9. Design, implement, test, and debug software using a high-level language that uses explicit pointers, as well as explicit static and dynamic memory allocation.
10. Work effectively in teams.
11. Communicate effectively in writing.

## Description of Assessment Tools

Quizzes/Homework/In-class exercises – Topics will be assessed using either an online or in-class quiz, homework assignment or an in-class exercise.

Team project – A team project in which the students implement a feature of the OS (file management system, process management system, memory managament system, scheduler etc) in C.

- Design and test plans phase - This phase will involve documenting the design and test process. Each team will submit a report.
- Implementation phase - The team will then implement the project in C.

Students will typically work on this project over a period of 5-6 weeks.

Small-scale projects – Projects in which students, working either individually or in teams, implement some other (different from the topic covered in the team project) operating system components in C (e.g. synchronization, scheduling, memory management). Students write a report for each project.

Exams – 2 or 3 (includes a final) typically 2 hour exam.

# Course Assessment Matrix

| | **Objective** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **Quizzes** | X | X | X | X | X | X | X | X | X | | | |
| **Team Project - Design and Test Plan** | | | | | | | | | | | X | X |
| **Team Project - Implementation Phase** | | | | | | | | | | X | X | |
| **Small-scale Projects** | | | | | | | | | | X | | X |
| **Exams** | X | X | X | X | X | X | X | X | X | | | |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

# Objective Evaluation

- The success criterion holds except for the following tool-objective pair:
  - Exam 1, Objective 7 (average grade was 63%)

# Summary of Relevant Student Feedback

- Introduction to C not thorough enough

# Instructors' Subjective Evaluation

- Grade for Objective 7 (process synchronization) on Exam 1 was low. However, the grade on the quiz was satisfactory. Students also did an in-class exercise on synchronization and the grade on this exercise was satisfactory.

# Recommendations for Next Offering

- Spend more time on introduction to C

-- ArchanaChidanandan - 03 Jun 2006

- - - - - Revision r1.1 - 03 Jun 2006 - 05:45 GMT - ArchanaChidanandan

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

# CourseAssessmentReportCSSE333IntroductionToDatabaseSystemsWinter200506

# Instructor(s)

Curt Clifton, Steve Chenoweth

# Learning Objectives

Students who successfully complete this course should be able to:

1. Describe how information models and systems affect and support the operations of organizations
2. Write and interpret Entity-Relationship models and Relational models
3. Justify other data modeling paradigms (such as legacy systems, object-oriented databases, etc.)
4. Design relational databases from problem statements
5. Demonstrate the basic aspects of physical database design
6. Employ SQL-based query languages to access and modify data
7. Develop programs using database techniques such as stored procedures, triggers, etc.
8. Argue the appropriateness of information privacy, integrity, security, and preservation methods in various situations
9. Discuss the technical and social issues involved in Online Analytical Processing (OLAP) systems and Data Warehousing

# Description of Assessment Tools

Daily Quizzes - short (5-10 questions) quizzes completed during class

Homework Assignments – pencil and paper exercises

Labs - 1 to 2 hour exercises using an industrial database running on a database management system

Project - individual or team project in design, documentation, implementation, and maintenance of a database

Exams – two two-hour exams

# Course Assessment Matrix

| | **Objective** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Quizzes** | X | | X | | | | | | |
| **Homework** | | X | X | X | X | | | | X |
| **Labs** | | | | | | X | X | | |
| **Project** | | | | | X | X | X | X | |
| **Exams** | X | X | | X | | | | X | X |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

# Objective Evaluation

The success criterion holds for all tool-objective pairs.

# Summary of Relevant Student Feedback

- Students very much appreciated the lab portion of the course.
- Students liked having the opportunity to select their own term projects and were very enthusiastic about this.
- Students disliked the homework problems, finding them disconnected from the other course material and confusing.
- Students found the two period lectures early in the course, before labs began, to be hard to follow.

# Instructors' Subjective Evaluation

- We are very satisfied with the students' learning and experience in the course and think that only incremental improvements are needed.

# Recommendations for Next Offering

- More hands-on activities should be introduced in the first three weeks of the course.
- Work load should be more evenly distributed between the labs. The stored procedure and JDBC labs, in particularly, were very time consuming. Because the version of SQL Server that we have used is being phased out, we recommend moving to SQL Server 2005 for the next offering.
- The new edition of the textbook has revised homework problems. These should be carefully reviewed to see that student concerns have been addressed. If not, a different text or newly designed homework problems should used.
- We might consider dropping relational algebra from the course and introducing additional practical concepts such as XML and an open-source database like MySQL. Ultimately, relational algebra is just another notation. While it is a more elegant notation than SQL, SQL is an accepted standard and includes all the concepts from relational algebra.

-- CurtClifton - 23 Apr 2006 - Weakened recommendation on Relational Algebra from "should consider" to "might consider"

-- CurtClifton - 21 Apr 2006 - Initial draft

----- Revision r1.2 - 23 Apr 2006 - 18:43 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

**CourseAssessmentReportCSSE372SoftwareProjectManagementWinter200506**

# CSSE 372 Software Project Management

# Course Assessment Report

# May 15, 2006

## Instructor

Don Bagert (two sections)

## Recommendations from Previous Offering and Corresponding Changes

- Adding more reading would be helpful (done)
- Course project for this course and CSSE 371 Software Requirements and Specification should be changed to a "real" project (done)
- More material should be added to this course and CSSE 371 (done)

## Other Significant Changes from Previous Offering

- None

## Learning Outcomes

Students who complete this course should be able to:

1. Explain several different lifecycle models for software engineering
2. Explain the fundamentals of software project management, including organizational structures and roles
3. Prepare a software project plan, including risk analysis
4. Lead or participate on a software project team, including sharing of responsibilities, building consensus, developing plans of action, and providing constructive feedback to peers
5. Plan, facilitate and participate in effective meetings
6. Participate as a member of a team that successfully monitors and controls a software project using traditional metrics
7. Conduct a software project retrospective
8. Explain the role of software engineering ethics and professional practices within the global society
9. Explain the role of motivation, satisfaction and lifelong learning in software professional careers
10. Explain standard configuration management methods
11. Apply their understanding of configuration management in a software project team

## Description of Assessment Tools

Homework Assignments – problems related to software engineering and software project management

Project - team project continued from CSSE 371 in which the students plan and manage a software project during its development

Exams – three one-hour exams

## Course Assessment Matrix

| Assessment Tool | Outcome | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| **Homework** | | X | X | X | X | X | | X | X | X | X |
| **Project** | X | | X | X | X | X | X | | | X | X |
| **Exams** | X | X | | | | | X | X | X | | |

# Objective Evaluation

The success criterion holds for all tool-outcome pairs.

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

# Summary of Relevant Student Feedback

- The students liked the "real" projects with the RHIT clients.

# Instructors' Subjective Evaluation

- I am satisfied with the students' learning and experience in the course.

# Recommendations for Next Offering

- None.

-- DonBagert - 15 May 2006

----- Revision r1.2 - 31 May 2006 - 14:35 GMT - DonBagert

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

**CourseAssessmentReportCSSE374SoftwareArchitectureAndDesignWinter200506**

# CSSE 374 Software Architecture & Design

# Course Assessment Report

# DRAFT

## Instructor

Mark Ardis

## Recommendations from Previous Offering and Corresponding Changes

- (none)

## Other Significant Changes from Previous Offering

- I tried to get students to take more responsibility for meeting with one another in later stages of the project. (This failed.)

## Learning Objectives

Students who complete this course should be able to:

1. Work effectively with a team of software project stakeholders, including customers and members of the development team.
2. Create problem statements from informal requirements.
3. Recognize the differences between problems and solutions and resolve their interactions.
4. Use fundamental design methods and strategies in the creation of a software architecture document.
5. Understand key decision-making activities necessary when architecting and partitioning software for any system.
6. Recognize issues in managing architectural attributes, such as quality, security, performance, and availability.
7. Evaluate designs for feasibility and soundness.

## Description of Assessment Tools

Homework - weekly exercises on course material

Project(Customer Role) - With 1 or 2 other students, perform the role of customer for a 7-week project that produces a software architecture document and an executable framework, as well as several other deliverables

Project(Architect Role) - With 1 or 2 other students, perform the role of architect for a 7-week project that produces a software architecture document and an executable framework, as well as several other deliverables

Project(Developer Role) - With 1 or 2 other students, perform the role of developer for a 7-week project that produces a software architecture document and an executable framework, as well as several other deliverables

Exams - two one-hour exams

## Course Assessment Matrix

| | **Objective** | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Homework** | | | | X | | | |
| **Project(Customer Role)** | X | X | | | | | X |
| **Project(Architect Role)** | X | X | X | X | X | X | |
| **Project(Developer Role)** | X | | X | | | | X |
| **Exams** | | | | | X | X | |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

## Objective Evaluation

The success criterion holds for all tool-objective pairs.

## Summary of Relevant Student Feedback

- Students like the test-first approach used this term.

## Instructors' Subjective Evaluation

- I am satisfied with the students' learning and experience in the course.

## Recommendations for Next Offering

- More examples of some of the notations would be good. A couple running examples would be even better.
- Conducting in-class sessions for the later stages of the project would have worked better than letting the students meet on their own. In other words, revert to the way Steve taught it last time.

-- MarkArdis - 06 Mar 2006

- - - - - Revision r1.1 - 15 Mar 2006 - 12:33 GMT - DonBagert

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

**CourseAssessmentReportCSSE451AdvancedComputerGraphicsWinter200506**

# CSSE 451 Advanced Computer Graphics

# Course Assessment Report

# Proposed – March 7, 2006

## Instructor

Cary Laxer

## Recommendations from Previous Offering and Corresponding Changes

- Keep the same ordering of deliverables. (accomplished)
- Add another lecture or two on some additional advanced graphics topic. (provided addtional material on radiosity this year)

## Other Significant Changes from Previous Offering

- None

## Learning Objectives

Students who complete this course should be able to:

1. Implement a ray tracer.
2. Read, understand, and discuss current professional journal articles in computer graphics.
3. Research and learn a graphics API (other than OpenGL?) and develop software that generates realistic images using that graphics API.

## Description of Assessment Tools

### Projects

One three week project in which students write a ray tracer using the OpenGL? graphics API and one five week project in which students implement a large computer graphics project of their own design in a graphics API other than OpenGL? that they researched. Both projects are done by teams of 3-4 students.

### Oral presentations

One 25-minute presentation on a current professional journal article the student has read, two 50-minute presentations on a graphics API the student has researched, and one 25-minute presentation on the researched graphics API project. The first presentation is done individually; the latter presentations are done with the student's team members.

### Written reports

One short (2-3 pages) report on the current professional journal article that the student read (written individually), one short (4-5 pages) users guide for the ray tracer (written as a team) and one long (20+ pages) report on the graphics API the student

researched (written as a team).

### Course Assessment Matrix

| | Objective | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| **Projects** | X | | X |
| **Oral presentations** | | X | X |
| **Written reports** | X | X | X |

### Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn passing grades in the course, the minimum grade on the portions of the assessment tools that are relevant to the learning objective is in the passing range.

## Objective Evaluation

- The success criterion holds for all tool-objective pairs.

## Summary of Relevant Student Feedback

- Students enjoyed the research nature of the course and the ability to help choose what they studied.
- Students also enjoyed the project nature of the course, having to implement two major projects.

## Instructors' Subjective Evaluation

- The course was very successful. I have used this ordering of work for three years now and it's been readily accepted by students all times.

## Recommendations for Next Offering

- Increase direction on the API project.
- Perhaps one or two mini-projects on other topics (but in this course that may be difficult due to the nature of the material)

-- CaryLaxer - 07 Mar 2006

- - - - - Revision r1.2 - 28 Mar 2006 - 12:45 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

# CourseAssessmentReportCSSE474TheoryOfComputationWinter200506

**CSSE 474 - Theory of Computation**
**Rose-Hulman Institute of Technology**
**Computer Science and Software Engineering Department**
**Course Assessment Plan**

### Instructor

Michael Wollowski

### Recommendations from Previous Offering and Corresponding Changes

None, first time assessed.

### Other Significant Changes from Previous Offering

None, first time assessed.

### Course Outcomes

Students who successfully complete this course should be able to:

1. Discuss the concept of finite state machines.
2. Design a deterministic finite-state machine to accept a specified language.
3. Explain context-free grammars.
4. Explain at least one algorithm for both top-down and bottom-up parsing.
5. Explain the Church-Turing thesis and its significance.
6. Determine a language's location in the Chomsky hierarchy (regular sets, context-free, context-sensitive, and recursively enumerable languages).
7. Prove that a language is in a specified class and that it is not in the next lower class.
8. Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs.
9. Explain how some problems have no algorithmic solution.
10. Provide examples that illustrate the concept of uncomputability.
11. Define the classes P and NP.
12. Explain the significance of NP-completeness.
13. Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it.

### Description of Assessment Tools

- Exams: Two exams and a final exam
- Assignments: Ten homework assignments

### Course Assessment Matrix

| Tool | Objective | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Exams | x | x | x | x | x | x | x | x | x | x | x | x | |
| Assignments | x | x | x | x | x | x | x | x | x | x | x | x | |

**Success Criteria**

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

**Objective Evaluation**

The success criterion holds for all tool-outcome pairs except for: Exams 3, 4, 5, 6, 8, and 12 and Assignments: 1, 3, 4, 5, 6, 10, 11, and 12. They were not assessed. The success criterion for the tool-outcome pairs for Exams: 1 and 2 are unknown as I ommitted to collect the appropriate data.

The list of course outcomes is too specific. It feels like a laundry list of particular items and does not capture the major themes of the course. Predominantly missing are outcomes more centrally related to Turing Machines. I strongly recommend that the list be revised and shortened.

**Summary of Relevant Student Feedback**

- More feedback through assignments. The assignments can have fewer problems on them, if they are more frequent.
- Students generally liked the instructor, the book, and the subject.
- Some students would have liked to see more practical applications than the one's presented in class.

**Instructors' Subjective Evaluation**

I taught this course for the first time, employing the textbook used by a colleague in the Mathematics Department. This explains some of the discrepancy between what was covered and the list of course outcomes.

I felt that the course went well.

It soon became obvious that I was not able to assign and appropriately grade as many assignments as I would have liked to and as should be assigned in a course like this. I feel that the proofs are best graded by the faculty teaching this course, however, due to the fact that I had two preparations with a total of 75 students, this was a new preparation, and that I had 48 students in the two sections of this course, there was simply not sufficient time to grade more assignments. Additionally, there were no qualified lab assistants available that could do a quality job when it comes to grading proofs. I talked to a colleague from the Mathematics department about grading so many proofs to find out that they only have few courses where this happens and that those courses are very time consuming.

**Recommendations for Next Offering**

- Assign more homeworks.
- Consider making the two sections of this course the only load of the person teaching it.

-- MichaelWollowski - 02 Jun 2006

- - - - - Revision r1.1 - 02 Jun 2006 - 15:18 GMT - MichaelWollowski

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE120FundamentalsofSoftwareDevelopmentISpring200506**

# CSSE 120 Fundamentals of Software Development I

# Course Assessment Report

## Instructors

Lisa Kaczmarczyk

Curt Clifton

## Recommendations from Previous Offering and Corresponding Changes

- The class needs more individual accountability. *This was provided by requiring more individual work on labs and by re-instituting a series of individual interviews with students.*
- The class needs to impart a sense of accomplishment to the students early on - with coding. *We introduced some simple exercises early in the course in which students experimented with language features, in particular definition, assignment, and method calls. The experiments required the students to predict the effect of a change in a program, make the change, determine the actual effect, and reflect on their prediction.*
- The students should get to write their own simple code early on to acheive this sense of accomplishment. *See response to previous point.*
- The overall immersion approach to this class should be changed, to reduce the appearance of "busy work" but more importantly, to reduce the "overwhelmed" feeling, and improve learning. *No change was made in this regard.*
- Use Big Java as the primary textbook and continue working to integrate appropriate readings into the course. *Done.*
- Do not use the online Stein textbook. *Done.*

## Other Significant Changes from Previous Offering

- Subversion was re-introduced to the course both as a mechanism for course management and for student collaboration.
- Students were allowed to choose their own Capstone project, rather than having every team do the same project.
- Because of scheduling conflicts, two small sections--one of 4 and one of 6 students--were offered. The small section sizes allowed the instructors to provide a tremendous amount of one-on-one instruction.

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Analyze and explain the behavior of simple programs involving:
    a. Fundamental non-object-oriented programming constructs, including:
        - Basic syntax and semantics of a higher-level language
        - Variables, types,expressions, and assignment
        - Simple I/O
        - Conditional and iterative control structures
        - Functions and parameter passing
        - Arrays
    b. Fundamental object-oriented programming constructs, including:
        - Basic syntax and semantics of a class
        - Fields,constructors and methods
        - The distinction between primitive types/variables and non-primitive types/variables
        - The relationship between the static structure of the class and the dynamic structure of the instances of the class (e.g. via Java's *new* operator)
        - Visibility modifiers (e.g. Java's *public*, *protected* and *private*)and mutability modifiers (e.g. Java's *final*)
        - The distinction between a static method/field and a non-static method/field
        - Implementing an interface
        - Extending a class
        - Responding to an exception
2. Design, implement, debug and informally test simple programs in an object-oriented programming language, including programs that:

    - Use the above fundamental programming constructs
    - Support a graphical user interface developed by using a GUI toolkit
    - Respond to user events
    - Use threads
    - Include "is-a" relationships among objects using a class hierarchy and inheritance
    - Use large-scale API packages
3. Apply decomposition techniques, including:
    - Using structured (functional) decomposition (breaking a function/program into subfunctions)
    - Encapsulating behaviors and data into classes
    - Using inheritance
4. Explain techniques that scale up to larger software development projects (and explain why they scale up), including:
    - Pair programming
    - Structured documentation comments
    - Implementing by using stubs
    - Implementing by using an iterative enhancement plan
    - Using code conventions (e.g., Sun's code conventions)
    - Decomposition techniques listed above
    - Using an integrated development environment
    - Using UML class diagrams
    - Unit testing using a tool like JUnit
    - Using simple metrics to evaluate quality of code
    - Using simple team building and management techniques like brainstorming, roles and task lists
5. Work effectively as members of a small team

## Description of Assessment Tools

- Daily quizzes which collectively assess various course objectives on an individual basis.
- Each of the major projects (e.g. Ball Worlds) has an online report which students complete as part of the project.
- Several projects and text reading assignments have written reports which students hand in individually.
- The reports have questions which collectively assess various course objectives.
- Mini project deliverables include:
    - From each team:
        - Problem statement, screen layouts and use cases that specify the project
        - An iterative enhancement plan of how to implement the project
        - A schedule by which to deliver the project
        - A task list that shows who is to do what, by when, and whether they did it
        - A UML class diagram for the project
        - Documented code produced in stages
        - A one page Executive Summary upon completion
    - From each student:
        - Discussion with Instructor of the feasibility of the Mini project
        - An assessment of her/his team's effectiveness
        - A reflection on what s/he learned from the Mini project
- Capstone project deliverables include:
    - From each team:
        - Problem statement, screen layouts and use cases that specify the project
        - An iterative enhancement plan of how to implement the project
        - A schedule by which to deliver the project
        - A task list that shows who is to do what, by when, and whether they did it
        - A UML class diagram for the project
        - Documented code produced in stages
        - A one page Executive Summary upon completion
        - A final presentation of their project, including a live demonstration
    - From each student:
        - An assessment of her/his team's effectiveness
        - A reflection on what s/he learned from the Capstone project

## Course Assessment Matrix

| Assessment Tool | Outcome | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Project reports | X | X | | | |
| Mini project discussion of feasiblity | | | | X | |

| | | | | | |
|---|---|---|---|---|---|
| Capstone project final presentations | X | | X | X | X |
| Capstone project reflection | | | | | X |
| Capstone project documented code | | X | X | | |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

## Objective Evaluation

The success criterion holds for every tool-outcome pair.

## Summary of Relevant Student Feedback

- A small number of students expressed concern that the immersion into object-oriented concepts and terms at the beginning of the course was "slightly confusing".

## Instructors' Subjective Evaluation

- The transition to Big Java as the primary text was successful. Having a text with more concrete examples seemed to ease students' concerns about the immersive nature of the course.
- Students and instructors benefitted from the extremely small section sizes this term. Because these small section sizes will not be the norm, it is difficult to draw strong conclusions about the overall course design.

## Recommendations for Next Offering

- Continue use of Subversion throughout the course. This was convenient for instructors and assistants in assessment and was very well received by the students.
- Look for additional opportunities to incorporate small, individually conducted experiments in the course. Ideally each programming topic would be:
  - introduced in class discussion,
  - investigated through experiments,
  - understood in lab exercises, and
  - applied in the Capstone project.
- Convert the self-check exercises from Big Java into Angel self-assessment exercises. This will allow students to check their own understanding and will encourage students to do the assigned readings. An assistant could be assigned to create the Angel exercises directly from the text.

-- CurtClifton - 03 Jun 2006 - Added sections to match standard CAR format. Converted as much HTML formatting as possible to TWiki formatting. Added my thoughts on recommendations and evaluation.

-- LisaKaczmarczyk - 27 May 2006

-- LarryMerkle - 19 May 2006 - Formatting changes

-- LisaKaczmarczyk - 03 May 2006

- - - - - Revision r1.2 - 03 Jun 2006 - 15:21 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE220FundamentalsofSoftwareDevelopmentIISpring200506**

# CSSE 220 - Fundamentals of Software Development II
# Rose-Hulman Institute of Technology
# Computer Science and Software Engineering Department

# Course Assessment Report
# Spring 2005-06

## Instructor

Don Bagert (1 section)

## Recommendations from Previous Offering and Corresponding Changes

None

## Other Significant Changes from Previous Offering

- Having the group project being an integrated effort from among all of the project teams, instead of each team doing the same project separately.

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Develop software that incorporates the following techniques:
    - Inheritance
    - Interfaces
    - Polymorphism
    - Casting
    - Function objects
    - Generics
    - Collections
2. Develop software in each of the following application areas:
    - Software that interacts with a database
    - Software that communictes over a network
    - Software that uses a graphical user interface
3. Perform the following steps of the software development cycle effectively:
    - Requirements elicitation
    - Design expressed as UML class diagrams
    - Implementation using an iterative enhancement plan and stubs/documentation before coding
    - Unit, system, and acceptance testing
4. Analyze the asymptotic worst, best, and average case run times of simple algorithms.
5. Select basic data structures (e.g, arrays, sequential lists, linked lists, stacks, queues, hash tables and trees) based on the time and space complexity of typical operations.
6. Develop software that uses recursive methods and data structures.

# Description of Assessment Tools

Written exercises: Short thought problems, mathematical analyses, or algorithm-design exercises.

Programming assignments: Several programming assignments of varying size and length, including a mini-project.

Projects: One large team project.

Exams: Two examinations, one at midterm and one at the next-to-last week (the last week was dedicated to finishing the project).

# Course Assessment Matrix

| | Outcome | | | | | |
|---|---|---|---|---|---|---|
| Tool | 1 | 2 | 3 | 4 | 5 | 6 |
| Written exercises | | | | X | X | |
| Programming assignments | X | X | X | | | X |
| Projects | | X | X | | | |
| Exams | X | | | X | X | X |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcomes pair selected above:

> Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcomes is in the proficient range.

# Objective Evaluation

The success criterion holds for all tool-outcome pairs, although outcome 3 for the group project was marginal. (Some of this was due to the difficulty and uniqueness of the project.)

# Summary of Relevant Student Feedback

- Some wish for less theory and more practice.
- Some suggested that the group project return to being the same project for all teams, and with each project team completely independent of the other.

# Instructors' Subjective Evaluation

- I am satisfied with the students' learning in the course.

# Recommendations for Next Offering

- An additional programming assignment near the end involving linked lists and/or binary search trees.
- The group project return to being the same project for all teams, and with each project team completely independent of the other.

-- DonBagert - 31 May 2006

- - - - - Revision r1.2 - 01 Jun 2006 - 18:54 GMT - MichaelWollowski

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE230FundamentalsofSoftwareDevelopmentIIISpring200506**

# CSSE 230 Fundamentals of Software Development III

# Course Assessment Report

## Spring 2005-2006

## Instructor(s)

Matt Boutell (1 section)

## Recommendations from Previous Offering and Corresponding Changes

- Make the workload more manageable by coordinating the due dates of programming and written assignments such that they don't occur on the same day. I made written assignments always due Monday and programming assignments always due Friday, thus making the workload more manageable and predictable.
- Add more justification for proof and induction. Students read, and we had a whole-class discussion on, an article entitled “Why Math” to justify studying proof and induction in Computer Science.
- Provide more examples of strong induction. I developed a resource comparing regular and strong induction that we used in class.

## Other Significant Changes from Previous Offering

- None

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Describe the classical data structures (list, stack, queue, tree, priority queue, hash table, graph, set, dictionary) and understand the issues involved in implementation choices for each.
2. Analyze and prove the asymptotic worst, best, and average case run times of algorithms, including non-trivial recursive algorithms, using Big O, little o, Omega, and Theta notation.
3. Analyze and prove lower-bound results on problems like comparison-based sorting and understand the implications of lower bound results.
4. Understand and analyze classical sorting, graph and tree-balancing algorithms.

## Description of Assessment Tools

Exams – Two exams and an optional final exam.

Programming assignments. Five to seven short (typically less than 500 lines of code each) programming assignments, at least two of them to be done individually, some by groups of two or three students.

Written Homework – Eight to twelve take-home assignments on various topics covered in class. An assignment typically consists of four to six problems from the textbook on various aspects of algorithms, data structures, and analysis.

## Course Assessment Matrix

| | **Outcome** | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| **Exams** | X | X | X | X |
| **Written Homework** | | X | X | |
| **Programming Assignments** | X | | | X |

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

## Objective Evaluation

The success criterion holds for all tool-outcome pairs except Written assignment 7 #6, used to evaluate outcome 3, for which the average was 63%.

## Summary of Relevant Student Feedback

- Most students liked revisiting familiar material in much greater depth. However, some want to be challenged more.
- Some students recognized the value of being held individually accountable (no partners allowed) on four of the seven programming assignments.
- A small contingency of students are vocal in their distaste for the theory and the written assignments that test their understanding of it. I acknowledge this, and will be upfront about it in the future. However, I don't plan to remove the theory, which for some will be their only opportunity to focus on it to this degree.
- Some students also found the Angel site difficult to navigate.

Noticeably absent were comments about an uneven workload, or about having enough examples of induction and proofs. These comments were prevalent last term, but I was able to address these issues this term.

## Instructors' Subjective Evaluation

Due to the time needed to develop another course from scratch this term, I was unable to devote as much energy to CSSE230 as I would have liked. However, I am pleased with the course because I think the students learned the material better than they did the previous term. I also feel that I successfully changed the course according to the recommendations from last term, as stated above, with two exceptions. First, I continued to use the in-class activites I developed last term to reinforce the theory, but did not develop any new ones. Second, while I think I did a better job motivating the mathematical aspects of the course, I think I could do more to target this motivation at the software engineers. One easy idea is to add a similar reading for math in software engineering, but I want to be aware of other ways to motivate the theory as well.

I did assign a previously-developed project, Markov, that used data structures from the Collections interface, as the first assignment. This helped the students review the introductory data structures they had seen in their previous coursework.

This is my first time teaching the course with a class composed mainly of students who took CSSE221 Honors. I feel that I will need to do a better job challenging these students in the future (typically Spring quarter). For example, the last programming assignment on encryption was much too easy.

## Recommendations for Next Offering

- Assign an article similar to "Why Math", but aimed at software engineers.
- Rework the final programming assignment to make it more challenging.
- Add an in-class assignment in which they code multiple algorithms to solve the same problem, but with different asymptotic complexity, to see first-hand the difference in runtime on a large data set. One example is to explore the different ways to concatenate strings in Java.
- Add at least one assignment that is less constrained than the current set of assignments, so as to require more creativity. Some examples:

1. One in which students are free to choose the most appropriate data structure for the problem at hand (and are graded on how well they can justify their choice).
2. One in which students compete to develop the most efficient algorithm to solve a difficult problem, such as efficiently calculating distances from a single point to a set of points on a grid.
3. One in which students need to implement their own data structure, or a hybrid of existing structures, to meet certain complexity requirements. One student started work on a linkedArrayList this term; there should be room to improve it.

-- MatthewBoutell - 02 Jun 2006 - Many small changes in wording.

-- LarryMerkle - 01 Jun 2006 - "objective" -> "outcome"

-- MatthewBoutell - 01 Jun 2006 - Added evaluative comments and recommendations once student feedback was available.

-- MatthewBoutell - 27 May 2006 - Initial post.

- - - - - Revision r1.4 - 02 Jun 2006 - 14:06 GMT - MatthewBoutell

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**

# CourseAssessmentReportCSSE304ProgrammingLanguageConceptsSpring200506

## CSSE 304: Programming Language Concepts

## Course Assessment Report
## Spring 2005-6

### Instructor

Michael Wollowski

### Recommendations from Previous Offering and Corresponding Changes

- Eliminate the coverage of PROLOG.
- Provide a more in-depth coverage of the interpreter materials.
- More clearly stated assignment.
- Avoid that few students monopolize the classtime when it comes to answering questions posed by the instructor.

### Other Significant Changes from Previous Offering

I made the changes listed above.

### Course Outcomes

Students who successfully complete this course should be able to:

1. Design, implement, debug, and test complex programs in a functional programming language
2. Explain the role of:
    a. Syntax specification
    b. Syntax and semantics
    c. Environments and continuations
    d. Types
    e. Abstraction
3. Explain the process of interpreting a program
4. Explain the relationship between syntax specification and implementation
5. Design, implement, debug, and test a complex interpreter

### Description of Assessment Tools

Programming assignments: About ten substantial programming assignments.

Exams: Two midterm exams and one final exam.

### Course Assessment Matrix

| | Objective | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| Programming assignments | X | X | X | X | X |
| Exams | X | X | X | X | X |

### Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above: Among

the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

**Ojective Evaluation**

The success criterion holds for all tool-outcome pairs.

**Summary of Relevant Student Feedback**

- Provide more detail for the programming assignments, so that they match up better with the test-cases. Students felt guessing at which constraints their programs may or may not have satisfied.
- Cut down on the earlier assignments, seemed like busy-work.
- Assignments and lectures reinforced each other well.
- Workload is way too high.

**Instructor's Subjective Evaluation**

I am very happy with the learning that students displayed. Students worked very hard and it was a pleasure to see them wrestle with the materials and to gain a solid understanding of some rather hard concepts.

**Recommendations for Next Offering**

- Consider starting the interpreter assignments about a week earlier, so that there is more time for the conversion to continuation passing style. By ninth week of the spring quarter, students have limited energy to absorb new materials.
- Consider providing more detail for the programming assignments so that students do not have to guess so much about which special cases their programs fail.
- Consider adding more written homeworks, especially when it comes to environments and conversions to CPS.

-- MichaelWollowski - 01 Jun 2006

- - - - - Revision r1.1 - 01 Jun 2006 - 19:38 GMT - MichaelWollowski

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE325FractalsAndChaoticDynamicalSystemsSpring200506**

# CSSE 325 Fractals and Chaotic Dynammical Systems

# Course Assessment Report

# Proposed – June 1, 2006

## Instructor

Cary Laxer

## Recommendations from Previous Offering and Corresponding Changes

- None; first time being assessed

## Other Significant Changes from Previous Offering

- None

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Determine the dimension of a fractal object
2. Determine the Hausdorff and Lebesgue measures of various sets
3. Use iterated function systems and L-systems to model fractals
4. Perform graphical analysis of dynamical systems
5. Control chaos
6. Determine mathematical properties of the Mandelbrot and Julia sets
7. Generate fractal images via computer software

## Description of Assessment Tools

### Projects

Several computer programming projects that allow students to implement algorithms that describe various types of fractals and that allow students to better understand the underlying mathematics of chaotic dynamical systems.

### Homework

Short exercises students complete covering concepts relating to chaos and fractals.

### Class work

In class exercises students work on to reinforce concepts covered in lecture.

### Course Assessment Matrix

| | **Objective** | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Projects** | | | X | | X | | X |
| **Homework** | X | X | | X | | X | |
| **Class work** | X | X | X | X | X | X | X |

### Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn passing grades in the course, the minimum grade on the portions of the assessment tools that are relevant to the learning objective is in the passing range.

## Objective Evaluation

- The success criterion holds for all tool-objective pairs except for the following:
  - Projects-Objective 3: One student did not turn in this project and thus received a 0 on it.
  - Homework-Objective 1: One student did not turn in this assignment and thus reveived a 0 on it.
  - Homework-Objective 2: One student did not turn in this assignment and thus reveived a 0 on it.

## Summary of Relevant Student Feedback

- Students really enjoyed the class and learned a lot about fractals and chaotic dynamical systems.

## Instructors' Subjective Evaluation

- The course was very successful. I was pleased to read one student's comment that said he now looks at clouds and sees fractal shapes.

## Recommendations for Next Offering

- Give some suggestions for different types of fractals to look into for the final project so students are not doing some of the same things that were already done in class just with a spin on it.

-- CaryLaxer - 01 Jun 2006

- - - - - Revision r1.1 - 01 Jun 2006 - 14:05 GMT - CaryLaxer

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE404CompilerConstructionSpring200506**

# CSSE 404 Compiler Construction

# Course Assessment Report

## PROPOSED: June 3, 2006

## Instructor(s)

Curt Clifton (1 section)

## Recommendations from Previous Offering and Corresponding Changes

- None

## Other Significant Changes from Previous Offering

- Theory of Computation was added as a prerequisite for the course. This let the instructor cover the formal foundations of lexical analysis and parsing as a review instead of as new material.
- The textbook was changed to Cooper and Torczon's *Engineering a Compiler*, which has a deeper treatment of the optimization and backend issues involved in modern compiler construction.
- Homework problems were suggested, not required. This change was intended to give students more control over their workloads and to give them some experience with a course format that they are likely to experience in graduate school.
- "Test-first teaching" using daily quizzes was introduced.
- The course project was revised and the description was rewritten.
- Project teams were reduced from 3-4 students to 1-2 students.
- Advanced features presentations were directly tied to the student projects in that the students had to describe the necessary changes to their compiler to implement the features (though there was not time to actually implement the features).

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Implement a program to tokenize an input stream based on a regular language definition
2. Convert a left-recursive context-free grammar into a right-recursive one
3. Construct the canonical set of LR(1) items and derive the tables used by an LR(1) parser for a given context-free grammar
4. Implement, using a parser-generator such as ANTLR and CUP, a program to parse a token stream and construct an intermediate representation based on a context-free grammar
5. Implement code to perform context-sensitive analysis and type-checking on a program encoded in some intermediate representation, such as an abstract syntax tree
6. Implement code to construct symbol tables for a simple, imperative programming language
7. Implement a program that takes code written in a high-level language and generates corresponding code in a

low-level target language, such as MIPS or Java Bytecode
8. Clearly explain selected, more advanced programming language implementation issues (such as optimization or adding non-main-stream features), based on independent study, and extend a working compiler to address them

# Description of Assessment Tools

Exam – One take-home exam.

Programming project - A significant term programming project completed by students working alone or in pairs. Students must implement a complete compiler with type checking for a small object-oriented language.

Project demonstrations – A demonstration of the programming project to the instructor intended to demonstrate understanding of the implemented compiler

In-class presentations - A short (approx. 15 minute) presentation about some language extension or compiler optimization focused on teaching the other students about the topic, including the syntax and semantics of the extension and the implementation techniques needed.

Daily quizzes - Short, in-class quizzes

# Course Assessment Matrix

| | **Outcome** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **Exams** | | X | X | | | | | |
| **Programming project** | X | | | X | X | X | X | |
| **Project demonstrations** | X | | | X | X | X | X | X |
| **In-class presentations** | | | | | | | | X |
| **Daily quizzes** | | X | X | | | | | |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

# Quantitative Evaluation

The success criterion hold for all tool-outcome pairs with the following exception:

- time did not permit actual implementation of the advanced programming language issues in outcome 8. Students were only required to describe the necessary changes to their compilers, though several actually did make the changes.

# Summary of Relevant Student Feedback

- Students considered the project to be crucial to the course.
- Some students thought that even less time should be spent on the formal background on lexical analysis and parsing (outcomes 2 and 3). This would allow a more in-depth treatment of optimization.
- Students reported that the workload was very high. None asked that it be reduced, but they suggested being even more forthright about the amount of effort involved.

# Instructor's Subjective Evaluation

The course was effective in achieving the learning outcomes.

I made adjustments to the daily quizzes throughout the first half of the term before settling on the right level of difficulty.

Students had a tendency to procrastinate on their project work. Because the milestones were due approximately every second week, their workload was "bursty".

Because this was the first year of the Theory of Computation prerequisite, I signed 3 students (of 12) into the class without the prerequisite. This required a more intensive review of the material in outcomes 2 and 3.

# Recommendations for Next Offering

- Strictly enforce the Theory of Computation prerequisite.
- Have students implement pieces of a small parser-generator and use it to satisfy outcomes 2 and 3.
- Schedule more frequent milestones. The following is a suggested schedule:
    - Weeks 1–2: finish parser-generator implementation
    - Week 2: select a lexer-generator (or hand-roll) and demonstrate scanning of a very simple language
    - Week 3: complete scanner
    - Week 4: select a parser generator and parse a very simple language
    - Week 5: complete parsing, AST generation, and pretty-printing
    - Week 6: typecheck declarations only, no method bodies
    - Week 7: complete typechecking
    - Week 8: code generation for a simple program that just prints out a 0.
    - Week 9: code generation for simple programs that don't use OO features
    - Week 10: complete code generation, add extension or optimization

# Revision History

-- CurtClifton - 03 Jun 2006 - Created

- - - - - Revision r1.1 - 03 Jun 2006 - 16:08 GMT - CurtClifton

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE432ComputerNetworksSpring20052006**

# CSSE 432 Computer Networks

# Course Assessment Report

## Instructor(s)

Archana Chidanandan (one section)

## Recommendations from Previous Offering and Corresponding Changes

- None.

## Other Significant Changes from Previous Offering

- None.

## Learning Outcomes

Students who successfully complete this course should:

1. Understand vocabulary of computer networking.
2. Understand networking services and how they are provided by multiple layers of protocols.
3. Understand principles and mechanisms of congestion control in networks.
4. Understand principles and mechanisms of routing in networks.
5. Understand security mechanisms for networks and why they are needed.
6. Be able to implement reliable file transfer services on an unreliable network.

## Description of Assessment Tools

Quizzes - In-class quizzes

Homework – Topics will be assessed using homework assignments

Small-scale projects – Projects in which students, working individually, implement computer networks protocol components in C

Final project - Project in which students, working in teams, implement computer networks protocol components

Exams – two typically one hour exams.

## Course Assessment Matrix

| | Objective | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| **Quizzes** | X | | | | | |
| **Homework** | | | X | X | X | |
| **Team Project** | | X | | | | X |
| **Small-scale Projects** | | | | | | X |
| **Exams** | X | X | X | X | X | |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

# Objective Evaluation

- The success criterion holds except for the following tool-outcome pair:
  - Homework, Outcome 3 (average grade was 63%)

# Summary of Relevant Student Feedback

Students asked for:

- A tutorial on using UNIX
- Discussion of current trends

# Instructors' Subjective Evaluation

- Grade for Outcome 3 (congestion control) on HW 3 was low. A few in-class examples may help with understanding the concept better.

# Recommendations for Next Offering

- Provide students with a tutorial on UNIX.
- Consider using Linux.
- Add more examples in class on topics such as congestion control.
- Have students research current trends and make presentations.

-- MatthewBoutell - 05 Jun 2006 - Minor tweaks.

-- ArchanaChidanandan - 03 Jun 2006

- - - - - Revision r1.4 - 05 Jun 2006 - 13:52 GMT - ArchanaChidanandan

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE461ComputerVisionSpring200405**

# CSSE 461 - Computer Vision
# Rose-Hulman Institute of Technology
# Computer Science and Software Engineering Department

# Course Assessment Report

## Instructor

J.P. Mellor

## Course Description

An introduction to 3D computer vision techniques. Both theory and practical applications will be covered. Major topics include image features, camera calibration, stereopsis, motion, and shape from x.

## Recommendations from Previous Offering and Corresponding Changes

First time assessed - no recommendations from previous offering.

## Other Significant Changes from Previous Offering

None.

## Course Objectives

Students who successfully complete the course should:

1. Understand the properties of projective lines, projective planes, and projective spaces.
2. Be able to use homographies to transformations image data (e.g. remove perspective distortion).
3. Understand the pin-hole camera model and projection matrices.
4. Understand epipolar geometry.
5. Be able to calibrate a camera.

6. Be able to reconstruct a 3D scene from image data.

# Description of Assessment Tools

Quizzes: Short (10 minute) unannounced true/false, multiple choice, and short answer quizzes.

Problem Sets: Seven or eight assignments including a combination of paper and pencil problems and programming assignments.

Take home tests: One or two take home tests. Take home tests are individual efforts.

Final Project: The final project will include a written reflection.

# Course Assessment Matrix

| | Objective | | | | | |
|---|---|---|---|---|---|---|
| Tool | 1 | 2 | 3 | 4 | 5 | 6 |
| Quizzes | X | | X | X | | |
| Problem Sets | | 2 | | | 2 | X |
| Take Home Test(s) | X | | X | X | | |
| Final Project | | | | | | X |

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

> Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range

# Objective Evaluation

The success criterion holds except for the following tool-objective pairs:

- Take Home Test (Q 5), Objective 3 (average grade was 66.7%)
- Problem Set (PS 5), Objective 6 (average grade was 67.9%)

# Summary of Relevant Student Feedback

- The course contained too much math.
- Review material for linear algebra would be helpful.
- Smaller, more frequent problem sets might be better.

- Good balance between theory and application.
- Problem sets complimented/reinforced course material well.
- More examples.

# Instructors' Subjective Evaluation

- Take home test question 5 regarding objective 3 may indicate weaknesses in understanding the pin hole camera model. The wording of the question also may have caused some confusion as a few individuals appear to have misunderstood the question (though they did not ask for clarification).
- Problem set 5 regarding object 6 does not indicate a problem. Two individuals failed to turn in this assignment and received a zero. Eliminating these two scores, the success criteria is met.

# Recommendations for Next Offering

- Split some of the larger problem sets into smaller pieces.
- Create a linear algebra tutorial.

---

*J.P. Mellor 2005-06-02*

-- MatthewBoutell - 29 Mar 2006 Deleted broken html link to image -- JpMellor - 02 Jun 2005

- - - - - Revision r1.3 - 29 Mar 2006 - 18:17 GMT - MatthewBoutell

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE490ImageRecognitionSpring200506**

# CSSE 490 Image Recognition

# Course Assessment Report

## Proposed May 27, 2006

## Instructor

Matthew Boutell (1 section)

## Recommendations from Previous Offering and Corresponding Changes

- None; this was the first time the course was offered.

## Other Significant Changes from Previous Offering

- None.

## Learning Outcomes

Students who successfully complete this course should be able to:

1. Describe morphological operations and use them to aid object recognition.
2. Describe and implement edgel detection.
3. Describe regions by using various shape representations (chain codes, curvature, moment invariants, spatial covariance matrix).
4. Decribe texture representations of image regions (statistics, co-occurrence).
5. Describe principles of classification: feature space, decision rules, decision surface.
6. Apply various classifiers such as neural networks and support vector machines to identify images and regions.
7. Describe and implement a clustering algorithm and apply it to image segmentation.

## Description of Assessment Tools

### Daily quizzes

In class exercises students work on to reinforce concepts covered in lecture.

### Exam

One in-class midterm exam covering feature extraction and classification concepts.

### Labs

4-6 Labs allowing the student to experiment with the concepts learned in lecture.

### Projects

2-3 computer programming projects with writeups that allow students to implement a full-fledged classification system (feature detection and classification) for images or regions.

### Course Assessment Matrix

| | **Objective** | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **Daily quizzes** | | | | X | | | X |
| **Exam** | X | X | X | X | X | | |
| **Labs** | | X | X | | X | X | X |
| **Projects** | X | | | | | X | |

### Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn passing grades in the course, the minimum grade on the portions of the assessment tools that are relevant to the learning outcome is in the passing range.

## Objective Evaluation

No data was collected, because I was too busy in my first time teaching this course to do so.

## Summary of Relevant Student Feedback

- The students really liked the course, as it was very different than any other course they had taken in the department. They thought the material was interesting, the labs helpful to reinforce the concepts, and the assignments challenging. They thought the overall amount of work was appropriate.
- They had several recommendations (noted below) that will help improve the course as it moves towards becoming a regular offering.

## Instructors' Subjective Evaluation

- I was happy to offer this elective to our students for the first time. I think that I was able to lay a good foundation of lecture materials, labs, and assignments this term, while having a number of ideas to improve the course next year. I agree with the students feedback, both positive and negative.

# Recommendations for Next Offering

- Add a unit on texture features, since students were ill-prepared to use them during their projects.
- Add a lecture on voting techniques, such as Hough transforms, to increase breadth.
- Add material on interpreting results from classifiers, e.g., determining which features the classifier is using to distinguish the classes.

Those reiterated by students:

- Refine the labs (too long and not enough direction on later ones).
- As a first assignment, have them walk through a Matlab tutorial at an optional lab session. This will allow us to get up to speed on Matlab without taking valuable class time on non-imaging topics.
- Give special-topics lectures germane to the projects during week 8. This will also serve to help keep me current in the literature.
- Assign the term project in week 4 (a week earlier) so they can have more time to research the literature.
- Add another project using non-color features (edges, regions, or texture). One possibility is to extend the fruit-finder to use edges. Another is to have them find circles by implementing a Hough transform?

-- MatthewBoutell - 31 May 2006 - Added evaluation and recommendation sections after student evaluations became available.

-- MatthewBoutell - 27 May 2006 - Created page.

- - - - - Revision r1.2 - 31 May 2006 - 20:15 GMT - MatthewBoutell

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE490RoboticsAndTeamworkSpring200506**

# CSSE 490, Teamwork and Robotics

# Course Assessment Report

## Proposed June 3, 2006

## Instructors

Steve Chenoweth (1 section)

David Mutchler (1 section)

## Recommendations from Previous Offering and Corresponding Changes

Not applicable -- this is the first Course Assessment Report for this course.

## Other Significant Changes from Previous Offering

Not applicable -- this is the first Course Assessment Report for this course.

## General description of this course

This is a "topics" (CSSE 490) course. It may someday become a regular course, although probably not in exactly the same format as it stands currently.

This course is unlike our other courses -- it is a ***service learning course***. Because it is unlike our other courses:

1. We have included this general description of the course along with the Learning Outcomes (below) and Assessment Plan (below).
2. The Learning Outcomes reflect our vision of the course and are not (yet) as measurable as they will eventually need t be.
3. The Assessment Plan is designed to obtain high-quality feedback easily. This feedback will help shape revisions of tl course design. Since the course is in its infancy, we concentrated on easily-obtained, high-quality feedback, rather than focusing on acquiring feedback that refers directly to the (still evolving) Learning Outcomes.

### Course Description

Here is a Course Description that might someday be in the online Undergraduate Bulletin:

> The mission of this course is to use service learning to help students internalize and deepen their understanding of how to work effectively in teams, while at the same time helping students improve their communication skills, their problem-solving, and their technical knowledge of robotics.

### Prerequisites

The *only* prerequiste is an interest in learning about robotics and teamwork by doing service. In particular, you must have a interest in ***helping middle and high school students*** learn about robotics and teamwork.

### Activities

To accomplish the course's mission, students will complete 3 team projects:

**Project 1: Design and implement a robot that could compete in the 2006 Beyond Botball competition described at http://www.botball.org/current-season/beyond_botball.php**

- Students will build the robots using Legos, Handyboard and Mindstorms controllers, and many types of motors and sensors. Some teams may use other, even more sophisticated, robotics parts.

**Project 2: Develop and deliver interactive exercises that teach middle-school and high-school students how to work effectively in teams.**

- In doing so, students will internalize their own understanding of how to work effectively in teams (per the goals liste above) and then deepen that understanding by developing effective ways to communicate that understanding to middle-school and high-school students.
- The best design for such exercises would allow the exercises to be extended easily to other groups of people, e.g. first-year Rose-Hulman students.
- The exercises will be delivered to middle-school and high-school students competing in the 2006 Botball competitio described at http://www.botball.org. The Rose-Hulman students will deliver the exercises, help teach the exercises to any middle-school and high-school teachers involved, and mentor behavior implied by the exercises in working with the middle-school and high-school students.

**Project 3: Investigate a technical area of robotics and present, orally or in writing, the technical ideas that the students learned from the investigation.**

- Each student team will choose their own area, but the ideas learned must extend (not repeat) ideas that the student team brings to this course.

# Learning Outcomes

Students who successfully complete the course should be able to demonstrate:

1. An understanding of how to work effectively in teams, including (but not limited to):
   a. The importance of creativity and how to combat blocks to creativity.
   b. The importance of specifications and how to express them.
   c. The roles of team members and how to conduct effective team meetings using those roles.
   d. How to reach consensus and how to handle conflicts.
   e. How to manage the progress of the team using task lists and other techniques.
   f. How to develop alternative designs and choose among them.
2. The goals of written, oral and interactive communication and how to implement them effectively.
3. An awareness of how professionalism impacts their career and how to maintain professional standards in their work.
4. An increased understanding of one or more technical subareas of robotics, e.g. the design and use of sensors, software development of concurrent code, path planning, vision, or programming language design.
5. An ability to help others understand what engineering is and why it is fun.
6. The ability to integrate the above mentioned course goals in the successful completion of three course projects.

Because this course is still in its infancy, the above Learning Outcomes reflect our vision of the course and are not (yet) as

measurable as they will eventually need to be. Here are some CommentsOnTeamworkAndRobotics that will be addressed in the next rendition of this course.

# Description of Assessment Tools

- ***Daily Reflection and Time Log survey***

Each weekday, you (each student) should complete a short *Daily Reflection and Time Log* survey. The survey for Day blah lets you log the work you did for this class on that day and reflect upon it. Complete the survey for every day, even those days on which you did no work for this class.

Each day's survey contains the same questions, as follows:

> **How much time did you spend on the date listed above doing work for Robotics and Teamwork?**
>
> Remember, count all your time -- attending class, completing this survey, thinking, reading, writing, building, programming, working with your Botball students, writing to them, traveling to them, etc.

> **If you responded 0 (zero) to the above question, skip all remaining questions.**

> **In a sentence or two, summarize what you did during the time you listed in the previous question.**

> **In a paragraph or two, reflect upon what you did during the time you described in the previous question.** For example:
>
> - What problems did you encounter? How did you address them?
> - How did you tell whether or not you were successful in the work you did? Did you get immediate feedback? Or will you get feedback later? Or ...?
> - If you worked with teammates, how did you organize your work and interactions? What strategies did you use for doing so?
> - What is something interesting that you learned or accomplished? Why is it interesting?
> - Are there any "take-away" lessons from your work? That is, any ideas you learned that would apply to other situations than the situation that you faced?

- In-class progress reports

Each team meets with the instructor about once a week to deliver an oral progress report on the activities that they have recently completed, or are in the midst of, or are planning.

- Research presentation

The presentation that students produce as Project 3, described above.

- Feedback from Botball students and teachers

Each Botball team should give feedback to the instructors regarding the interactions between that team and their Rose-Hulman mentors. The feedback is in two forms: normative evaluation to help the Rose-Hulman mentors improve and summative evaluation that the instructors will use to measure success on the learning outcomes.

In practice, we did not do this form of feedback this term.

- Anonymous course and instructor evaluations

Students should complete both the standard Rose-Hulman course evaluation and an evaluation that we tailor for this course.

# Course Assessment Matrix

| | **Outcome** | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| **Daily Reflection and Time Log survey** | X | X | X | X | X | X |
| **In-class progress reports** | X | | X | | X | X |
| **Research presentation** | | X | | X | | |
| **Feedback from Botball students and teachers** | X | X | X | | X | |
| **Anonymous course and instructor evaluations** | | | | | | |

This course is in its infancy and this is its first Course Assessment Plan. We recognize that the above may be "too much assessment" for purposes of a Course Assessment Plan, but we have included it all so that the designers of next year's Course Assessment Plan can build upon this work.

# Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

(Note from David Mutchler: I think that this course needs a stronger success criteria in future years.)

# Quantitative Evaluation

We did not obtain Feedback from Botball students and teachers this year. Also, the In-class Progress Reports were measured only informally. Finally, while students completed a Daily Reflection and Time Log survey for each day of the course, we used it as an assessment tool only by computing the total hours of effort students reported.

That said, the course was fully successful -- EVERY student earned proficient grades on every measured item.

Again, because this course is still in its infancy, the above Learning Outcomes and their assessment reflect our vision of the course and are not (yet) in the form that they will eventually need to be.

The students actual grades were determined by the following (announced in the Syllabus at the beginning of the course):

> The learning outcomes for this course are listed above. Because this is a ***service-learning*** course, we will ***assess your effort*** rather than directly measuring your success at the learning outcomes.

To that end, you will record your work in this course on a daily basis, by using Angel to complete a short survey each weekday. Your daily surveys will determine your grade as follows:

About 12 hours per week of work for this course (counting *all* your work, in class and out) earns you an A, assuming that:

- You work steadily -- you should work about 12 hours or more most weeks, not merely average 12 hours of work per week.
- You distribute your work reasonably among the three projects, thereby succeeding in all three projects.
- You provide high-quality reflections with your daily surveys and you provide a high-quality report from your Project 3 research.
- You attend class regularly.

About 8 hours per week of work for this course (counting all your work, in class and out) earns you a B, under the same assumptions as above.

We were very pleased with this novel grading plan. It helped students see their effort and reflect upon it -- that is the most appropriate result one could hope for from grades in a service-learning course.

Also, it was easy to measure effort, allowing us to focus our efforts on feedback to students and improving the course.

In practice, we lowered the hours-per-week requirements at the end of the course to 10 hours per week for an A and 7 hours per week for a B. These were more in line with student expectations and also reflected the fact that students tend to underreport effort. The following example helped us see that 12 hours per week (a standard for many professors) was the wrong cutoff point for an A -- a particular student who graduated from Rose-Hulman with a 4.0 average and who worked appropriately in this course reported only 9.8 hours per week as of week 9.

## Summary of Relevant Student Feedback

- "The team building and brainstroming exercises were good and I've tucked them away for future use."
- "This course was focus on teamwork, however my team was slightly dysfunctional and moreover the ultimate goal of mentoring teams (as a learning experience) was lost on the geographic inaccessability of the teams we were to mentor."
- "I'm not really sure how much I learned. It was a service learning course, so this isn't too surprising in some respects. Most of my learning was hands-on, which is probably a good thing. Unfortunately, beyond the first few weeks of the course, the amount of learning was more or less the same as if I were doing it as an extracurricular activity, since the course became sort of guided "free time". It was helpful to have the TA and both professors there, but very little of the time was spent learning any material. The paper at the end is good, but it would be better if it had started a bit earlier."
- "It'd be nice if we learned a little more about robots."
- "Great class and learned a lot."
- "This course was very effective in both getting our team to work together and work with both the middle school and high school teams."
- "I learned a lot about teamwork , I learned a lot of good things as well as a lot of bad things."
- "It was very interresting trying to communicate with our teams... put quite the real world spin on things. I liked that we did not have class every day (since we did not have things to talk about every day)."
- "This course was both hands on in Teamwork and Robotics. Rather than just hearing about how things are we were given a little direction and told to go after it."
- "It is an interesting class that is a good way to learn things. The lab setting helps keep students interested, this was the easiest class to stay awake in that I've had so far at Rose."

## Instructors' Subjective Evaluation

Overall comment: This course was a huge success for almost all of the students. Students gained important experiences that other Rose-Hulman courses provide in only limited ways, including:

- real-world communication experiences
- very explicit presentation and practice of teaming techniques
- a huge amount of practice at working in teams
- quickly learning new ideas (from Robotics) that they present to clients (the Botball students)
- the value of learning by providing service

Students ***averaged*** reporting 10 hours per week on the course, with some students working up to 14 hours per week -- all the more impressive since students typically underreport effort. Since this is an experiential course, simply putting in those hours is important.

We presented teamming techniques about once a week. These techniques were very well-received by the students. The students applied the techniques successfully in the direct followup exercises that we provided. All of the students definitely learned many new teaming techniques. Because the teaching of these techniques was experiential, we expect that students will retain the techniques for a long time.

Students did not always do as good a job at applying the techniques to their own work -- either with Botball teams (Project 1) or in their own Beyond Botball team (Project 2) -- as we had hoped. They did well in specific exercises that we asked them to do, but they generally did not continue to use the techniques throughout their work.

The Rose-Hulman students who had direct contact with the Botball teams had a wonderful experience -- it provided important practice at communication skills as well as letting the Rose-Hulman students internalize their teaming lessons. We witnessed some of these interactions first-hand and we were VERY impressed. The students prepared for the sessions, delivered them professionally, and discussed Plus/Deltas afterwards. The students definitely grew in their ability to communicate effectively, their understanding of professionalism, and their desire to be good citizens and ambassadors for their discipline.

While all the Rose-Hulman student teams had direct contact with Botball teams, not every student on every team had significant direct contact.

The Beyond Botball experience was much more successful this year than last year (in the course's first offering). Students tried applying Extreme Programming techniques to working as a team in the Beyond Botball competition. This experience was very valuable for them, in that it showed both the strengths and the weaknesses of Extreme Programming techniques.

Despite working much more effectively in their Beyond Botball teams, the robots that they produced were largely nonfunctional (but much better than last year's robots). Students did a poor job of estimating effort and time management.

The Research Presentations varied from quite good to barely passable.

In sum, the service-learning was a huge success for almost all the students. We want to make it a success for ALL the students --- see recommendations.

# Recommendations for Next Offering

1. Keep the three projects. They provide the right balance of experiences. All three already work well and will work even better with small changes.
2. Continue to make sure that the course is available to all years (including freshman) and is advertised to all disciplines. The mix of students is important to its success.
3. Continue to tie the timing of the course to Botball (or a similar competition).
4. Keep the grading scheme (but with levels of about 10, 8, and 6 hours for A, B, C, respectively). Add a requirement that each student includes direct (but possibly long-distance) contact with Botball teams as part of their personal experience of Project 1.
5. Insist that students do the Daily Reflection and Time Log for Day X no more than a day or two after Day X. Encourage this by giving feedback on their reflections within a day or two of the reflection being entered. Manage this feedback by hiring a student assistant to provide it (preferably a student who has taken the course). Automate the process of determining who has not done their Daily Reflection and Time Log and sending them email reminders. Catch anyone who falls behind early and help them see that passing the course requires completing the Daily

Reflection and Time Logs. Automate the process of moving the grade per the effort-level recorded in the Daily Reflection and Time Log to Angel's Gradebook; make that process happen daily so that students always know their grade.

6. Develop more structure to help students communicate with teams via distance-learning techniques. Hire an assistant to make this happen. There are many resources available -- phone, internet phone, instant messenger, email, video conferencing, to name a few. Some of the Rose-Hulman teams made good use this year of some of these resources, but the students need to be trained how to use these resources and we need a more clear infrastructure for the resources.
7. Currently, the Daily Reflection and Time Logs ask the student to report effort for each day -- good. It then asks students to explain what they did and to reflect upon the experience -- good. But we need a way to capture each student's effort on subtasks -- the three projects and their direct (but possibly long-distance) contact with Botball teams. Accomplish this by adding appropriate questions to the survey with numerical answers, e.g. "How many hours did you spend working on Project 1? Project 2? Project 3? How much of your Project 1 was direct contact with Botball teams?" Automate the process of capturing that effort and recording it in Angel's Gradebook.
8. Add to the Syllabus an explanation that Project 3 is a pair project in which students interact with each other (instead of just splitting the work). Start Project 3 earlier in the course.
9. Keep the excellent exercises that Steve designed. Make sure that they are put in a Resources section of Angel for use in this course and elsewhere. Develop more such exercises and integrate them into student's work even more.
10. Find some way to encourage students to apply what they learned about teamwork in ALL their work in this course. (Not sure how to do this. Maybe just close attention to students' reflections??)
11. Insist that students do a Project Plan as part of Beyond Botball. Provide careful feedback on that plan and help students with estimating effort (do some exercises??) for Beyond Botball and managing their time for it (do some exercises??)
12. Do a study based on the Daily Reflection and Time Logs and repeat that study next year. Report results in SIGCSE or elsewhere.
13. Continue to have Steve and David teach this course (possibly with others too). Steve is a world expert at teaching teamwork -- this course is a natural for him and makes terrific use of his skills. David has the tie to Botball that enables the service-learning component. (Other competitions could be used, of course, but it requires a tremendous effort to make the connections between a course and Botball or its like. Hence, stick with Botball as long as there is someone like David that is willing to volunteer that effort.)
14. Find some means for students in the course to keep the lab uncluttered.
15. Have the course listed as meeting MTR with one of those days being 2 periods. This provides a time when teams can work together for longer than a single period. Publicize to students the fact that students can take the course even if the "extra" hour conflicts with another course -- as long as only a few students are in this situation, it will not cause problems. Also, the instructors need not attend the "extra" period.
16. ***Showcase this course.*** For example: Work with the Rose-Hulman's Public Relations and Admissions staff to help publicize the course -- locally (e.g. newspaper articles), to prospective students (e.g. have them visit), to parents (e.g develop a brochure about the course), and to the world (e.g through a world-class web site). Ask notables like the Terre Haute mayor and Rose-Hulman's president to visit the class and preside over festivities like the Beyond Botball competition. Involve faculty from other departments -- guest lectures on topics in robotics (e.g. mechanical issues) and teaming (e.g. how teams work in their discipline). Bring in guest lecturers from around the world -- experts on robotics and experts on teamwork. Publish the idea of the course and studies on its effectiveness in conferences and journals. Create a Botball Board that is beautiful to look at and has a pleasant permanent location, with robots on it that visitors can interact with.

-- DavidMutchler - 08 Jun 2006 - Added the standard sections for information from previous CARs (filled in as "N/A", since there is no previous CAR for this course). Also Steve filled in the section for Student Feedback.

-- LarryMerkle - 05 Jun 2006 - "objective" -> "outcome"

-- DavidMutchler - 03 Jun 2006

- - - - - Revision r1.5 - 08 Jun 2006 - 14:02 GMT - DavidMutchler

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

# CSSE 490-01, Software Architecture and Design II

# Course Assessment Report

## Nov 9, 2005

## Instructor

Steve Chenoweth (one section)

## Recommendations from Previous Offering and Corresponding Changes

- This is the first time this course is offered.
- It includes and extends material previously in CSSE374, and becomes the second term of that course. See recommendations listed there.
- It includes a trial of teaching the introduction to Interaction Design (which thereafter is moved to CSSE371, with more advanced topics retained here).

## Learning Objectives

Students who complete this course should be able to:

1. Design and build effective human-computer interfaces using standard methods and criteria. [To be removed after Spring 2005, and instead put into CSSE 371.]
2. Understand the basic ingredients of successful software product lines – How to do multiple releases and versions of software.
3. Analyze the quality and economics of existing designs and systems. Includes make vs. buy decisions, and discussion of component selection.
4. Using UML and other methodologies and notations, create the overall design for a system and document this design. Includes methods to develop, prototype and document architectures.
5. Apply the process by which architectures get created, in terms of technologies, economics, people and processes. Includes detailed design from high level design, and development from software patterns and frameworks.

## Description of Assessment Tools

**Project** – This is the major student deliverable in the course, a full-term project done by teams of two peers (averaging 50% of the student's course time each week, but only about 25% of class time).

The project is chosen by each team as a continuation of work started in CSSE 374 the term before. In this project the students will go through all the remaining steps of architecture and design, which start with an existing software architecture document (from CSSE 374), and continue with various design activities including application of patterns and frameworks, and make-versus-buy decisions.

In the final weeks of the course, teams do a partial implementation of the system so as to validate their design. The term project tests the second objective, above, in that students actually build two partial versions of a system from their design (one in 374 and one in this course), and also because they build generality into their frameworks, with the idea of extending their design into multiple systems.

For 2004-5, the project included a substantial interaction design activity. For 2005-6, this activity will be shortened but not eliminated, assuming that the students had significant exposure to interaction design in CSSE 371. (See footnote about course sequencing.)

Students will present and demonstrate these projects during week 10 of the course. The primary goal of that activity is to assess how the actual implementation has tested their designs.

**Homeworks** – These assignments are primarily individual ones such as answering questions at the ends of the chapters and very small projects to test concept comprehension. This is a minor segment of the course designed to be a starter on new topics. The exception is the presentation of an architecture case study, counted as homework, which requires a team of two students to do a full-hour presentation on one of the case histories from Bass's Software Architecture book. This assignment is clearly an application of earlier learning.

**Biweekly quizzes** – These are four short essay quizzes of approximately 30 minutes duration. All are closed book, done in class. The quizzes test for broad knowledge and application of concepts. A sample question is, "Ch 8 of Bass says that 'the complexity of flight simulators grew exponentially' over a 30 year period. Why was that particularly the case for this application? How would you allow for an application that doubled in size periodically?"

## Course Assessment Matrix

| | **Objective** | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Team Project** | 2 | X | X | 2 | 2 |
| **Homeworks** | | | X | | |
| **Biweekly Quizzes** | | X | | | |

In the above table, "2" indicates that the objective will be assessed on two different times in the team project.

## Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

## Objective Evaluation

All objectives were met by all students successfully completing the course.

## Summary of Relevant Subjective Student Feedback

- "The material made sense and was reinforced by what we learned/are learning in other SE courses."
- "Learned a lot about interaction design and patterns. The project reinforced these concepts well."
- "I thought the ongoing project really helped the course, although forcing the design patterns was not always helpful for our particular project."
- "Great topics, very informative. The material covered real-world topics."

## Instructor's Objective and Subjective Evaluation

- Every student in this course achieved "A" level design work, even applying industry standards for software development. E.g., The architectural frameworks developed had to be provably extensible and use standard "patterns" to achieve this goal.
- These results probably are optimized by the fact that this time the course was taught to volunteers. Next year it will be a required course.

## Recommendations for Next Offering

- In the gap left by moving intro to Interaction Design, increase the material on software patterns, revisit refactoring, and provide an additional cycle of development and rearchitecting in the project.
- The course needs a book on patterns, and the fact that the Interaction Design book is now purchased for 371 could allow for a required text. Still looking for something beyond the GoF text, which is getting old.

## Footnotes

**Course sequencing** - Moving interaction design into CSSE 371 will make 371 another prerequisite for this course. We've avoided

course sequencing in the 37x series so far, but the fact is that 374 and 377 also assume students can write requirements and quality attribute scenarios. CSSE 377 also will make use of the texts from 371 as well as the two from 374.

-- SteveChenoweth - 09 Nov 2005

----- Revision r1.3 - 10 Nov 2005 - 16:52 GMT - LarryMerkle

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

**TWiki . CSSE .**
**CourseAssessmentReportCSSE497498499SeniorProject20052006**

# CSSE 497-498-499 Senior Project

# Course Assessment Report

# DRAFT – June 1, 2006

## Instructors

Mark Ardis and Steve Chenoweth

## Learning Outcomes

Students who complete this course should be able to:

## Communication Skills

- **Reading:** read technical documents and offer constructive criticism of their content and style
- **Writing:** write several different types of technical documents
- **Oral presentation:** prepare and deliver technical material at the appropriate level of detail

## Management Skills

- **Leadership:** lead a small software team (if team leader) or ability to support the leadership of the team (if not)
- **Time management:** estimate and monitor personal time across multiple tasks
- **Meeting facilitation:** lead and participate in small groups in constructive meetings
- **Estimating:** estimate effort required to complete technical tasks
- **Risk:** assess project risks and plan mitigation strategies
- **Planning:** prepare a feasible plan for the accomplishment of several technical tasks
- **Monitoring:** track the progress of several tasks according to a plan

## Technical Skills

- **Analysis:** analyze technical requirements and proposals for feasibility and to model the consequences of proposed solutions
- **Design:** construct appropriate abstractions of problems and solutions
- **Coding:** produce and inspect implementations of software according to project standards
- **Testing:** prepare test plans and to participate in both unit-level and system-level testing activities

# Professionalism

- **Ethics:** identify and prevent unethical professional behavior
- **Intellectual property issues:** make appropriate professional judgments regarding choice of methods for protecting intellectual property
- **Social issues:** evaluate and avoid possible negative social aspects of a software product
- **Relationships with clients:** interact with clients in a professional manner

# Course Assessment Matrix

| Outcome | Artifacts |
|---|---|
| Reading | Peer reviews of documents |
| Writing | Deliverables, especially documents |
| Oral presentation | 2 presentations/term |
| Leadership | Peer evaluations (1/term) |
| Time management | Peer evaluations (1/term), Project Plan |
| Meeting facilitation | Meeting Feedback, Meeting Minutes |
| Estimating | Project Plan |
| Risk | Risk Plan |
| Planning | Project Plan |
| Monitoring | Project Plan |
| Analysis | Requirements Specification |
| Design | Design Document |
| Coding | Delivered Code, Code Inspections |
| Testing | Test Plan, Test Results |
| Ethics | (none) |
| Intellectual property issues | Project Plan |
| Social issues | Requirements Specification |
| Relationships with clients | Client Feedback (1/term) |

# Success Criteria

The course will be considered fully successful if the following statement holds for every artifact-outcome pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

# Outcome Evaluation

All outcomes were met by all students successfully completing the 3-term course sequence. Those outcomes not measured by peer evaluations or by the client were judged by the two instructors, working together to resolve any questions of what standards to apply. Because the artifacts were submitted by teams, these also were graded on a team basis. The final presentation was a public poster presentaion/demo, and scores were

recorded from other CSSE faculty who went to each booth and graded them. These scores corroborated the passing evaluation judged by the two instructors.

Not every team submitted every artifact -- these were considered to be weakest in the area of test plans and test results. However, in almost all of these cases, the lack of an artifact was acceptable to the client. For example, in some cases the client was also a developer and cared only about delivered code.

# Summary of Relevant Subjective Student Feedback

- "I learned what its like to be working on a real software project."
- "It really helped me realize how difficult putting together a project can be. There are so many things that can go wrong at any one time, and most of those things can happen, so you need to be prepared for it."
- "The course provided real world experience which unfortunitly included some of the less than positivie aspects."
- "It gives a much better feeling as to the combined parts of software engineering courses and computer science courses than any other."
- "I learned a lot on my own and the hands-off approach allowed me to learn what I needed without impediments."
- "It was an excellent chance to apply the SE skills I learned in previous classes."
- "The project was too small for five people. It was completely unchallenging."
- "I got to work with technology I'd never seen. Also the grouping techniques which we used were new to me as well."

# Instructor's Objective and Subjective Evaluation

- Most of these projects were successful, in terms of delivering a desired product to their clients.
- The problem of sizing a project and team for one another (see next to last comment, above) is exacerbated by the fact teams self-select their projects. The instructors do agree to this choice. Some projects always turn out to be too easy and others too hard, of course.
- The essential problem of getting a project off to a fast launch often seems to be complicated by a less than desirable client responsiveness, particularly in terms of agreeing to a set of requirements that a team can consider "baselined." One team had the further startup problem of finding funding for the project; this made them among the slowest to begin development.
- In these 2005-6 senior projects we allowed teams to "roll their own process" in hopes of getting some to try agile methods. However, they also became less diligent about doing some of the documentation when this would have been a benefit (as when they need to be more formal about requirements with a remote client).

# Recommendations for Next Offering

- As one way of improving on the requirements management problem for next year, we are asking teams to pick a leader and a requirements engineer at the end of their junior year, and then begin client contacts over the summer. This also puts those contacts closer to when the clients submitted their project ideas.
- Next year we still want to encourage teams to try agile methods. For this coming year, we will emphasize more the need to also use documents to help manage the project. Toward that end, we provided next year's teams with a list of "must have" documents: problem statement and requirements, architecture and high level design, project plan including risk plan, and test plan. These required

documents fit closely with the course outcomes, above.
- In sizing teams for next year (2006-7), we added people from among those students who had not formed their own team. However, with one exception, we kept all these teams at around five students. So, we are still not sure if we got this part right! As before, we threw out a few proposals that were clearly too hard or too easy.

-- SteveChenoweth - 01 Jun 2006

----- Revision r1.2 - 01 Jun 2006 - 20:06 GMT - SteveChenoweth

Copyright © 1999-2003 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback.

# CASO RosE Portfolio Rating Report

**Annual Report**

Prepared for:

Cary Laxer
Computer Science & Software Engineering Department Head
Rose-Hulman Institute of Technology

Prepared by:

Julia Williams & Shannon Sexton
CASO Co-Facilitators
Rose-Hulman Institute of Technology

&

Timothy Chow
Office of Institutional Research, Planning and Assessment
Rose-Hulman Institute of Technology

July 8, 2005

**ROSE-HULMAN INSTITUTE OF TECHNOLOGY**
**OFFICE OF INSTITUTIONAL RESEARCH, PLANNING AND ASSESSMENT**
5500 WABASH AVENUE-CM11 TERRE HAUTE, INDIANA 47803-3999
TELEPHONE: (812) 877-8816 FAX: (812) 877-8931 EMAIL: CASO@rose-hulman.edu

OFFICE OF INSTITUTIONAL RESEARCH, PLANNING AND ASSESSMENT

# MEMORANDUM

**DATE:** July 8, 2005
**TO:** Cary Laxer, CSSE Department Head
**FROM:** Julia Williams, Shannon Sexton, and Timothy Chow
**SUBJECT:** 2005 RosE Portfolio Rating Report

The Office of Institutional Research, Planning and Assessment has prepared a report on the 2005 RosE Portfolio Rating Results. In June, a team of sixteen faculty served as portfolio raters of student submissions to the Communication, Ethics, and Teams Objectives. The results of their rating were compiled to show the overall performance of students and to show performance by individual departments. This information can be used for the purpose of preparing for ABET accreditation.

Thank you for providing us the opportunity to be of ongoing service to you and the faculty of Rose-Hulman.

**CASO** **RosE Portfolio Rating Report**

# Table of Contents


| | PAGE |
|---|---|
| **EXECUTIVE SUMMARY** | 1 |
| **FINDINGS** | 2 |
| **DISCUSSION** | |
| Ethics | 3 |
| Teams | 3 |
| Communication | 4 |
| Exemplary Documents | 4 |
| **RATIONALE** | |
| Purpose of Rating | 6 |
| Scope of Rating | 6 |
| **METHODOLOGY** | |
| RosE Portfolio Rating Process | 7 |
| Portfolio Usage Comparison | 7 |
| Statistical Analysis | 8 |
| Faculty Raters and Assigned Objective/Criterion | 8 |
| **IMPROVEMENTS** | |
| Improvements | 9 |
| CASO | 9 |
| Review of Culture and Global Objectives | 9 |
| Development and Management of Process | 9 |
| **APPENDICES** | |
| Appendices Directory | 10 |

# CASO RosE Portfolio Rating Report

## Executive Summary

The RosE Portfolio System (REPS) is an online portfolio system that allows Rose-Hulman students to make submissions to six Institute-wide learning outcomes: Communication, Ethics, Teams, Contemporary Issues, Culture, and Global. Student submissions to three learning outcomes are assessed each year, although collection of student documents to all six Institute learning outcomes occurs every quarter. This report presents the results of the 2005 Rating Session for the first outcomes set (Communication, Ethics, Teams). These data are also compared to results from the 2003 Rating Session, the last time that documents for these objectives were rated. The data are accompanied by an explanation of the findings and suggestions for improvements. There is an Glossary of terms used within the report in Appendix A.

In June 2005, sixteen faculty rated submissions to the RosE Portfolio under the Communication, Ethics, and Teams Objectives. Faculty raters use a Yes/No benchmark for one rating question: Does the submission meet the criterion at a level expected of student who will graduate from Rose-Hulman?" The 2005 RosE Portfolio Rating Session results were compared to the rating results obtained during the 2003 Rating Session, the last time that student submissions to Communication, Ethics, and Teams were assessed.

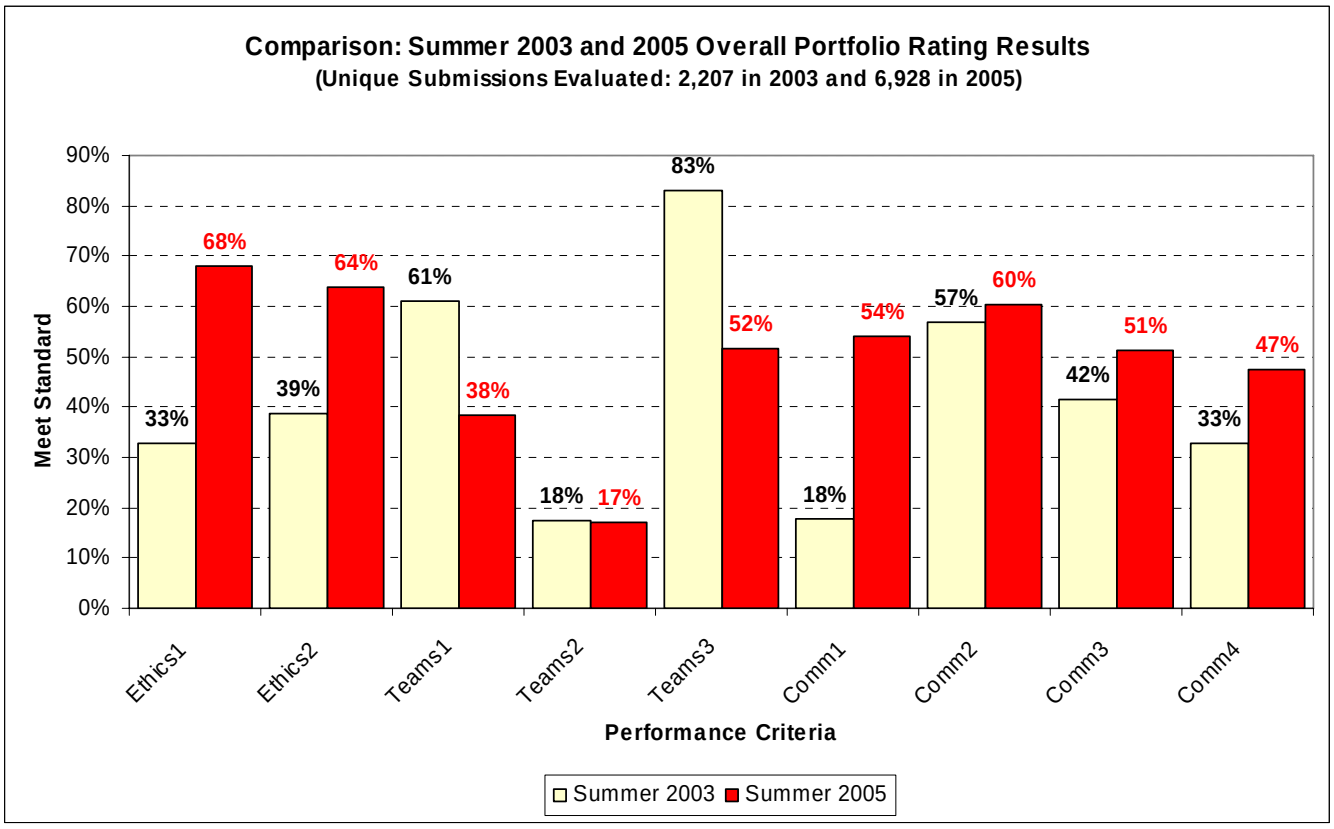
The results of the 2005 ratings reflect improvements in student performance in Communication and Ethics from the 2003 ratings. There were downturns in the results in Teams. Overall more students are making more submissions to REPS than before. In addition faculty are making a conscientious effort to insure that the assignments they create to use for portfolio submissions align properly with the learning outcome, performance criteria, and rubric. Faculty raters of portfolios made suggestions regarding types of documents that align well with the evaluation rubrics. These suggestions will help faculty in technical departments design assignments and encourage students to make submissions that will be successful in portfolio ratings.

Improvements to the RosE Portfolio Rating Process are scheduled for the 2005-06 academic year. The Commission on the Assessment of Student Outcomes (CASO) plans to conduct workshops for new and returning faculty. The purpose of the workshops is to improve rates of student performance during the next evaluation session and to assist faculty in becoming more familiar with the portfolio submission and rating process. CASO will also assist departments in their preparations for upcoming accreditation visits from ABET and the Higher Learning Commission.

## RosE Portfolio Rating Results

The Overall Rating Results produced in the 2005 Rating Session are provided below. The 2005 Results (in red) are compared to the results produced during the 2003 Rating Session (in yellow). "Overall" means the ratings for all files rated during the Rating Session. Rating results for individual departments are provided in the Appendices at the end of this report.

**Figure 1**
**Comparison of Overall Rating Results**
**% of submissions meeting the criteria**




Improvements in the student performance on the Ethics objective were impressive: up 35% for Criterion 1, and up 25% for Criterion 2, when compared to the 2003 rating session. The overall drop in student performance on the Teams Objective was an unexpected outcome of the 2005 Rating Session. Given the concerted effort made by faculty to design appropriate assignments and guide students' submissions to REPS, an improvement in performance was predicted. There were significant gains in students meeting the performance criteria for Communication. In Criterion 1, for example, the percentage of documents that were rated as acceptable tripled from the 2003 results. Similar improvements were seen in the other three criteria as well. A full discussion of the rating results is provided in the next section of the report.

## RosE Portfolio Rating Results

This section of the report provides context for understanding the results produced by the 2005 Rating Session. It also provides possible explanations for both the improvements and downturns in results for particular objectives and performance criteria. These explanations were developed in discussion with faculty raters, faculty, and department heads. Suggestions for improving student performance in the future are included in the Conclusions section following this discussion.

## Ethics

The reason for these gains lies in two improvements made during the Communication, Ethics, and Teams collection cycle. First, technical departments made a concerted effort to design assignments that provided students with opportunities to demonstrate the two criteria very specifically. For example, the Criterion 1 rubric requires that students quote from the professional code of ethics. Faculty from several technical departments reported making such references to a code of ethics an explicit outcome for their assignments. Second, faculty who participated in previous rating sessions were able to bring feedback to their departments that helped them design more appropriate assignments. Seeing firsthand the rating process and the kinds of documents that meet the standard helped departments make their Ethics assignments more suitable for meeting these criteria. The data do not indicate, however, that students must take a required ethics course in order to meet the Ethics criteria. The wording of the criteria, as well as the performance levels described by the rubrics, are suitable to courses where students are given the opportunity to reflect on the ethical dimensions of professional engineering practices. These criteria are often demonstrated through the senior capstone design courses.

## Teams

As noted in the Findings section, student documents submitted for the Teams objective in all three performance criteria did not meet the passing standard at the level expected. Comments collected from the faculty raters of the Teams Objective provide explanation of possible reasons for performance in this area:

1. As stated in the Rubric for Criterion 1 (See Appendix B for Objectives, Criteria, and Rubrics), the faculty raters looked for explicit reference to the "roles/tasks that each member [of the team] would perform on the project." A student who only described his/her own role on the team (such as in a resume) would not meet the rubric and would receive a "No" rating.

2. Meeting minutes, Gantt charts, and lab reports would meet the rubric as long as the tasks/roles were linked to individuals. Some Gantt charts did not list individual team members or assign them to specific tasks.

3. While many individual faculty are working to ensure that students submit appropriate documents, it may be the case that there are some faculty who are not monitoring student submissions as carefully. In the case of the Teams Objective, lack of faculty monitoring, which contributes to a higher percentage of "No" ratings, may be compounded by the fact that students make individual submissions, rather than a single team submission.

Faculty raters of the Teams 1 Criterion reported that they only applied the first part of the Rubric, "Share responsibilities and duties." The raters found that very few documents addressed the second part of the rubric.

This is a clear result of an assumption by some faculty who required their students to make submissions to Teams 1; they believed that the second part of the rubric had been removed after the matter was discussed by CASO. Unfortunately verification of the change in REPS was not made prior to the Rating Session by either these faculty or by the Rating Session Coordinator.

## Communication

The explanation for these improvements lies in the increased guidance that the Technical Communication faculty gave to students making submissions. Students were required to submit particular documents to REPS, and submissions were part of the course grade.

In addition, assignments that were more appropriate to the criteria were selected, and documents produced for those assignments were submitted to REPS. For example, for Criterion 1, students are asked to "identify readers/audience, assess their previous knowledge and information needs, and organize/design information to meet those needs." Students in RH 330 were required to submit their Summary Justification memos, a document in which they explained the process of writing multiple summaries of the same journal article for different reader types. In the Summary Justification memos, students explained their choices of technical terminology, focus, background, and other issues that made each summary appropriate for a different audience (such as an engineering manager reader, a high school student reader, an entry level engineer reader). These memos were brief documents, but they targeted the criterion explicitly.

The appropriateness of the assignment also explains the drop in performance for Communication Criterion 3: "Test readers/audience response to determine how well ideas have been relayed." During the 2003-04 academic year, students completed a Client Documentation Project that included Usability Testing of a set of instructions that the students wrote for a process, product, or procedure supplied by an on-campus client. Rating results were Yes 221, No 73. Because of a shift in course focus, RH 330 faculty decided not to continue with a particular assignment in the 2004-05 academic year. During the 2004-05 academic year, these client projects were not required, and a different assignment was used. Rating results declined to Yes 110, No 240. RH 330 faculty will review these results and determine if the client project should be reinstated or if a different assignment will serve equally well in the next document collection cycle, or if the performance criterion should be revised.

## Exemplary Documents and Examples of Suitable/Unsuitable Documents

During each rating sessions, portfolio raters are asked to identify documents that they find represent excellent work that effectively matches the performance criteria and the rating rubric (see Suitable/Unsuitable Documents table reference). These exemplary documents can be used by faculty in two ways. First, faculty may wish to see examples of documents and assignments that are appropriate to particular performance criteria. Second, faculty can use the exemplary documents and assignments as models on which to design their own assignments that they plan to require students to submit to REPS.

While CASO has discussed the possibility of making the exemplary documents available to students in REPS, we have not yet resolved the confidentiality issue with regard to student work. This concern will again be a part of the CASO agenda for the 2005-06 academic year.

Faculty raters this year were asked to provide summary comments about Exemplary Documents to the Rating Session Coordinator. From these summary comments, we have compiled a list that identifies example document types that are suitable for a particular Performance Criterion (see Figure 3). This list should help technical departments understand more clearly why some of their students' submissions did not receive an acceptable rating.

**Figure 3**
**Documents Deemed Suitable by Raters**
**2005 Rating Session**

| Suitability of E-Portfolio Documents for Rating | | |
|---|---|---|
| **Criteria** | **Examples of Suitable Documents** | **Examples of Unsuitable Documents** |
| Communication 1 | • Summary justifications (RH330) | • Rhetorical analysis of target forum (RH 330) |
| Communication 2 | • Trade journal paper (RH 330)<br>• Research paper from a HSS course<br>• Submissions with a bibliography worked the best | • Lab reports<br>• Single reference reports<br>• Analysis papers<br>• Final design project reports |
| Communication 3 | • Usability tests (RH 330) | • Facial expressions or body language<br>• Peer reflection from presentations (RH 330) |
| Communication 4 | • MS Word files<br>• PDF files<br>• Text documents<br>• RAR files | • Resumes<br>• PowerPoint presentations<br>• CAD files<br>• Maple or Matlab |
| * Teams 1 | • Meeting minutes if tasks linked to individuals<br>• Gantt charts if tasks linked to individuals<br>• Lab reports if tasks linked to individuals | • Product of a team's efforts<br>• Resume highlighting skills of team members |
| Teams 2 & 3 | • Charts in Word or Excel<br>• Pugh Charts<br>• Gantt Charts | |
| Ethics 1 | • Code of ethics is directly referred to in the submission<br>• Code of ethics should be explained & reflected on | • Code of ethics pasted into a submission |
| Ethics 2 | • Applications of ethical theory<br>• Discussions of professional codes of ethics<br>• Discussions of case studies<br>• Video Discussions<br>• Debate summaries | |
| All Criteria | | • CAD<br>• Solid edge drawings<br>• .cmbl files<br>• Maple |

* Suitable for first part of rubric only

## Purpose of Rating

The purpose of the RosE Portfolio Rating Session is to assess evidence of student learning in six non-technical Institute outcomes. These outcomes are grouped into two sets: Communication, Ethics, Teams; and Contemporary Issues, Culture, Global. Evidence of student learning in these six outcomes is collected each year through assignments made by faculty in technical and non-technical departments. For example, some engineering faculty require that students submit documents from capstone senior design courses as evidence for the Teams outcome. Humanities and Social Sciences faculty require that students submit documents produced in their courses for evidence of the Global and Culture outcomes. Definition, collection of documents, and assessment and evaluation of evidence for technical learning outcomes is the province of technical departments.

## Scope of Rating

Student documents are submitted to the RosE Portfolio in every quarter of every academic year. Rating of those documents occurs each summer immediately after the end of the spring quarter; however, rating focuses on alternating outcome sets each year. For the 2005 Rating Session, therefore, all submissions to Communication, Ethics, and Teams collected since 2003 were rated. The last rating session for these objectives was summer 2003. Table 1 illustrates the rating cycle. After the Rating Session results are compiled, they are used by technical departments for preparation of their own ABET reports, and by the academic dean and technical departments for their Higher Learning Commission accreditation reports, as well as reporting to constituencies interested in the evaluation and continuous improvement of RHIT. Contemporary Issues, Culture, and Global, the second set of outcomes, was rated in summer 2004 and will be rated again in summer 2006.

**Table 1**
**Portfolio Rating Cycle**

| Objective | 2003-04 | 2004-05 | 2005-06 | 2006-07 | 2007-08 | 2008-09 |
|---|---|---|---|---|---|---|
| Communications | ✓ | | ✓ | | ✓ | |
| Ethics | ✓ | | ✓ | | ✓ | |
| Teams | ✓ | | ✓ | | ✓ | |
| Contemporary Issues | | ✓ | | ✓ | | ✓ |
| Culture | | ✓ | | ✓ | | ✓ |
| Global | | ✓ | | ✓ | | ✓ |

The Commission on the Assessment of Student Outcomes (CASO) began development of the RosE Portfolio in 1996. CASO established two objectives in developing an electronic portfolio for the purpose of student learning outcomes assessment. First, CASO wished to create a data collection method that would allow students to submit documents they produced in technical and non-technical courses (authentic evidence), rather than using standardized tests, documents produced solely for the purposes of assessment, or surveys. Second, CASO wished to develop an online portfolio that would allow for submission, rating, and evaluation of evidence of student learning within a single electronic system. While the RosE Portfolio System (REPS) has been used since 1998, a new REPS software product was developed in the summer of 2004 in cooperation with Quest Information Systems, Inc. The 2005 Rating Session was the first time the new software had been used for a full rating session.

## RosE Portfolio Rating Process

Rating submissions to the RosE Portfolio has followed the same methodology since the system was initiated in 1998. Rose-Hulman faculty are hired as portfolio raters. Attempts are made to involve faculty from many different departments on campus to ensure objectivity in rating and broad based familiarity and participation in the process. Raters work for two days in the Dynamics Laboratory (Olin 203) and are compensated at the rate of $250 per day. The Rating Session Coordinator (Julia Williams) facilitates the process and assigns pairs of raters to rate student submissions for a particular Performance Criterion. For example, a mechanical engineering faculty member and a chemistry faculty member may work as a rating pair assessing the student files under Criterion 4 of the Communication Objective.

The rating process consists of three steps.

1. First, REPS requires that the faculty rater pair rate a set of three shared documents. The rating is made on the basis of a Rating Rubric and uses a Yes/No benchmark for a single rating question: "Does this submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman." The raters must agree in their rating of the initial document set. This step is called Inter-Rater Reliability (IRR). The process ensures that raters look for the same qualities and components in order to rate documents as acceptable.

2. Second, if the raters agree in their IRR, the system then allows them to proceed with a set of ten documents, each rater reading and rating a different set of ten. The system records their rating for each document. The system also introduces a shared file every twelve documents in order to check that the raters have maintained their Inter-Rater Reliability. Failure to rate the shared document identically will cause the system to stop the raters so that they can recalibrate their work before moving on to another document set, ensuring validation of the rating process.

3. Third, the raters can provide comments about the rating session or about the student submission in the Comment boxes. In addition to the work of rating, faculty raters are asked to identify Exemplary Documents among the files that they assess. Discussion of Exemplary Documents occurs on page 8 of this report.

For the Rating Session, portfolio raters use the Objectives, Performance Criteria, and Rubrics that have been developed by CASO based on feedback from faculty in all departments. A complete list of the Objectives, Performance Criteria, and Rubrics for Communication, Ethics, and Teams is provided in Appendix B.

## Portfolio Usage Comparison

In the 2003 Rating Session, 2,207 unique submissions were made to REPS. During the 2005 Rating Session, 6,928 unique submissions were made to REPS. This represents a 68% increase in usage of the system since 2003, the last time that submissions to the Communication, Ethics, and Teams objectives were rated. In 2003, only documents collected during one academic year were rated. In 2005, we rated documents that had been collected for two years, reflecting the Portfolio Rating Cycle (see Table 1).

## Statistical Analysis

Frequency data and descriptive statistics were used to report the findings of the rating sessions. An ANOVA was run to test the statistical significance of the difference between the results from 2003 to 2005. As such, all data is presented in terms of percentages and total numbers of students.

## Faculty Raters and Assigned Objective/Criterion

**Table 2**
**Faculty Rater by Objective**

| Objective | Criterion | Rater | Department |
|---|---|---|---|
| Ethics | 1 and 2 | Heinz Luegenbiehl | HSS |
| | 1 and 2 | Anneliese Watt* | HSS |
| Communication | 1 and 3 | Rebecca DeVasher | Chemistry |
| | 1 and 3 | Bill Weiner * | ABBE |
| | 2 | Glen Livesay | ABBE |
| | 2 | Don Richards* | ME |
| | 4 | Bruce Black | ECE |
| | 4 | Clark Merkel | ME |
| | 4 | Michael Mueller* | Chemistry |
| | 4 | Renee Rogge | ABBE |
| Teams | 1 | Mark Ardis* | CSSE |
| | 1 | Jim Hanson | CE |
| | 1 | Sudipa Kirtley* | PHOE |
| | 1 | Bill Kline | EM |
| | 2 and 3 | David Erwin* | Chemistry |
| | 2 and 3 | Richard House | HSS |

Faculty who have served as raters in previous Rating Sessions are marked with an "*".

## Improvements to the RosE Portfolio Rating Process

The 2005 REPS Rating Session provided important insights into the development of the portfolio assessment process. By completing the second rating cycle for the Communication, Ethics, and Teams Objectives, we have closed the loop on our assessment process and can apply what we have learned in order to improve the process. The following is a discussion of the work proposed for particular academic and academic support units that will improve the overall process.

## Commission on the Assessment of Student Outcomes (CASO)

CASO is responsible for the development and maintenance of REPS and the Institute-wide Assessment Plan. CASO will be engaged in the following activities during the 2004-05 academic year:

1. Review the 2005 Rating Results to evaluate student performance and make suggestions for improving rating results during future sessions.

2. Verify the current statement of the Objectives, Criteria, and Rubrics in REPS to ensure there is consistency between the system and the expectations of faculty

3. Conduct three workshops for faculty: Introduction to RosE Portfolio (fall, for new faculty); Using the RosE Portfolio in the Classroom (winter and spring, for faculty who will require that students make submissions to their portfolios)

4. Develop strategies for engaging more faculty in the portfolio process (through rating, feedback, etc.)

5. Develop strategies for helping faculty align student submissions with the Performance Criterion (perhaps in a workshop format

6. Update and redesign the CASO webpages to provide more effective resources for faculty and academic support units

## Review of Culture and Global Objectives in the HSS Department

The next assessment cycle will involve rating student submissions to the Contemporary Issues, Culture, and Global Objectives. When the HSS Department discussed results from the 2004 Rating Session, they expressed interest in refining the Performance Criteria for Culture and Global. This work will occur in the summer of 2005 in order to make changes to the criteria available in REPS before students begin to make submissions in the fall.

## Development and Management of Process at Departmental Level

Faculty in technical departments will probably wish to have focused feedback on their students' performance in the three objectives. They will want to know how they can improve their assignments, how they can encourage students to make good submissions, and how they can make the portfolio submission process efficient and effective. CASO and IRPA are committed to assisting departments in any way that they can.

## Appendices

A. Glossary ..... 11
B. RosE Portfolio Rating Rubrics ..... 12
C. Summer 2005 Overall Portfolio Rating Results Chart ..... 19
D. Summer 2005 Overall Portfolio Rating Results Table ..... 20
E. Summer 2005 Overall Portfolio Rating Results by Class ..... 21
F. Overall Comparison Table Summer 2003 vs. Summer 2005 ..... 22
G. CSSE Department Comparison Table Summer 2003 vs. Summer 2005 ..... 23
H. CSSE Department Comparison Chart Summer 2003 vs. Summer 2005 ..... 24
I. Overall Comparison Table by Class Summer 2003 vs. Summer 2005 ..... 25
J. Summer 2005 Portfolio Rating Results for CSSE Department ..... 26

**Appendix A**

## Glossary

**CASO:** Commission on the Assessment of Student Outcomes, the Institute-wide committee charged in 1996 with the development an Institute assessment plan . CASO is responsible for . . .

**CET:** Communication, Ethics, and Teams objectives, one of two objectives sets that are rated on a two-year basis. Contemporary Issues, Culture, and Global are the other objective set. Submissions to that set will be rated at the end of the 2005-06 academic year.

**Performance criterion:** Performance criteria are the measurable statements that define the learning objective.

**Rating rubric:** a brief description of the level of student performance in a document that meets the standard for the Rose Portfolio. Rubrics were developed by faculty serving on Commission on the Assessment of Student Outcomes (CASO) in consultation with members of academic departments on campus. Rubrics are reviewed and evaluated each year to maintain their currency. Rubrics are organized in three parts: Description, a brief statement that explains more specifically what is expected for a particular performance criterion; Example, a list of the types of documents that are appropriate for the particular criterion; and Not Acceptable, a description of documents that are not appropriate submissions.

**REPS:** Rose E-Portfolio System. REPS was developed beginning in 1996, and the old software was terminated in the fall of 2004 when it was replaced by the new REPS software. The new REPS software was inaugurated in the fall of 2004 and developed by the faculty and staff of Rose-Hulman, Ventures, and Quest is Information Systems, Indianapolis.

**Unique submission:** a submission that has been made to the portfolio for at least one performance criterion. Submissions to the portfolio may be mapped to more than one performance criterion, so the designation of “unique” gives a truer picture of the number of files submitted that must be rated.

**Appendix B**

# RosE Portfolio Rating Rubrics

**Objective: Ethics**
**—A recognition of ethical and professional responsibilities**

**Criterion 1: Demonstrate knowledge of a professional code of ethics.**

**Rubric:**
Description: The student should clearly establish that they have read and understand a professional code of ethics (e.g., by quoting a specific passage, referencing a specific verse, or otherwise displaying an intimate knowledge of the code).

Example: A passage or section from the code of an engineering professional organization (e.g., AICHE, NSPE, ASCE, etc.), could be cited, or one from another professional organization code (e.g., the American Psychological Association code could be cited by students in a psychology class).

Not Acceptable: Citing the RHIT student code of ethics, or other unrelated codes of conduct, like the Constitution, the Ten Commandments, or engineering codes and standards that deal with issues like details of construction, testing, or product specifications (e.g., ASME Pressure Vessel Code, ASTM specifications, building codes, or safety codes).

**Criterion 2: Evaluate the ethical dimensions of professional engineering, mathematical, and scientific practices.**

**Rubric:**
Description: The student should describe an application of a professional code of ethics related to engineering, mathematics, or science, with a clear connection between the code provision and the application.

Example: A case study, a personal experience, or a debate about a contemporary issue could be used as long as there is an ethical dimension involved.

Not Acceptable: Describing an ethical issue outside of engineering, mathematics, and science (such as social science) or an application that does not clearly involve ethics.

**Objective: Contemporary Issues**
**—An understanding of how contemporary issues shape and are shaped by mathematics, science, and engineering**

**Criterion 1:** Demonstrate an awareness of how the problem is affected by social concerns and trends.

**Rubric**
Description: The student should define the technical problem and demonstrate the link between it and social concerns/trends. A student should recognize the larger social context in which the problem emerges.

Example: A student may link the oil crisis to the development of hybrid cars.

Not Acceptable: A student describes a typical day in their life, there is no discussion of the larger social context, or the problem is not a technical problem.

**Criterion 2:** Demonstrate an awareness of how the proposed solution(s) will affect culture and the environment.

**Rubric**
Description: The student should offer a proposal for solving the problem posed in Criterion A. The link between the solution and the culture at large/environment must be explicit.

Example: The student could evaluate the hybrid car solution to the oil crisis by measuring pollution levels.

Not Acceptable**:** The cultural/environmental effects are not evaluated, or the problem is not a technical problem.

## Objective: Global

**Criterion 1:** Demonstrate an awareness of the historical development of cultures and societies.

**Rubric**
Description: The student should discuss how particular historical events impact selected societal/cultural elements.

Example: A student could examine the historical events (political alliances, treaties, wars, social attitudes) that have influenced US immigration policies.

Not Acceptable: A historical chronology without a discussion of the impact on societies/cultures.

**Criterion 2:** Show an awareness of the relationships of nations and the interdependence of peoples around the globe.

**Rubric**
Description: Student should show how the actions of one or more countries affect each other.

Example**:** A student could submit a paper that discusses international relations; papers that come from sociology, anthropology, geography, political science, economics, or history (when the analysis deals with more than one country).

Not Acceptable: Comparison of two forms of government without analysis of their affect on one another; papers that simply contrast two forms of government but do not show the interdependence of the two forms.

## Objective: Culture
## —An ability to understand diverse cultural and humanities traditions

**Criterion 1:** Perform, read, or otherwise engage in artistic, literary, and/or other forms of culture.

**Rubric**
Description: The student should document a non-technical, creative, or artistic expression of any culture.

**Criterion 2:** Recognize the importance of contributions of peoples from other cultures to the students' professions and/or personal lives.

**Rubric**
Description: Students should demonstrate how an event, tradition, culture, or ideology of one culture impacted the student's professional or personal perception of that culture in a thoroughly reflective manner.

**Criterion 3:** Evaluate an issue or problem from other cultural perspectives.

**Rubric**
Description: The student is able to analyze a problem from other perspectives.

## Objective: Teams

**Criterion 1: Share responsibilities and duties, and take on different roles when applicable.**

**Rubric**
Description: The student should describe the roles of each team member and indicate how sharing responsibilities facilitates project completion.

Example: In a team member evaluation, a student might list the tasks performed by each member of the group. A student might be team leader for one meeting, recorder for another meeting, responsible for generating AutoCad drawings, and check design calculations.

Not Acceptable: To list only the student's own responsibilities and only one role performed.

**Criterion 2: Analyze ideas objectively to discern feasible solutions and then build consensus.**

**Rubric**
Description: The student should indicate which ideas were considered and how the ideas were evaluated, discuss the outcome of the analysis of ideas with a brief justification, and articulate the process the group used for determining their final solution.

Example: A student might list pros and cons for multiple ideas, might justify the group's decision based on economic factors, and might submit minutes from the group meeting where the final solution was selected.

Not Acceptable: A list of ideas without analysis, selecting an idea without correlating it to some selection criteria, and submitting the final report without explanation of the process by which consensus was reached.

**Criterion 3: Develop a strategy for action.**

**Rubric**
Description: The student should identify the tasks necessary for successful completion of the group project.

Example: A student might use a Gantt chart to prepare a timetable for completion.

Not Acceptable: Failure to identify either tasks or milestones for completion.

## Objective: Communication
## —An ability to communicate effectively in oral, written, graphical, and visual forms

**Criterion 1: Identify readers/audience, assess their previous knowledge & info. needs, & organize/design information to meet the needs.**

**Rubric**
Description: The student should give a brief description of the readers/audience for the document or project, an assessment of their needs, and a demonstration of how the information design accommodates that audience.

Example: A student may include an audience analysis worksheet or a usability study of the audience. A student could draw the rater's attention to a glossary of terms included in the document in order to accommodate readers less familiar with the topic.

Not Acceptable: A statement of the intended audience without an analysis of their previous knowledge or information needs, or a statement that the student accommodated readers without reference to particular features of the document or project.

**Criterion 2: Provide content that is factually correct, supported with evidence, explained with sufficient detail, & properly documented.**

**Rubric**
Description: The student should verify the information provided with reference to sources/authorities and document those authorities in a style appropriate to his/her discipline (IEEE, APA, MLA). The student should also provide adequate detail to assist the readers' understanding of the topic.

Example: A student could submit a Technical Communications report that documents statements cited from authoritative sources and provides a bibliography of such sources.

Not Acceptable: Any document that offers inadequate information without attribution or any document that appears to be plagiarized from an authoritative source.

**Criterion 3: Test readers/audience response to determine how well ideas have been relayed.**

**Rubric**
Description: The student should discuss how she tested her document/project on readers/audience and how that testing determined revisions to the document.

Example: A student may submit the results of a usability study that demonstrates how effectively the document met her readers' needs for information and discuss what changes were made to the document based on the results of the study.

Not Acceptable: No evidence of reader/audience testing of the document/project.

**Criterion 4: Submit work with a minimum of errors in spelling, punctuation, grammar, and usage.**

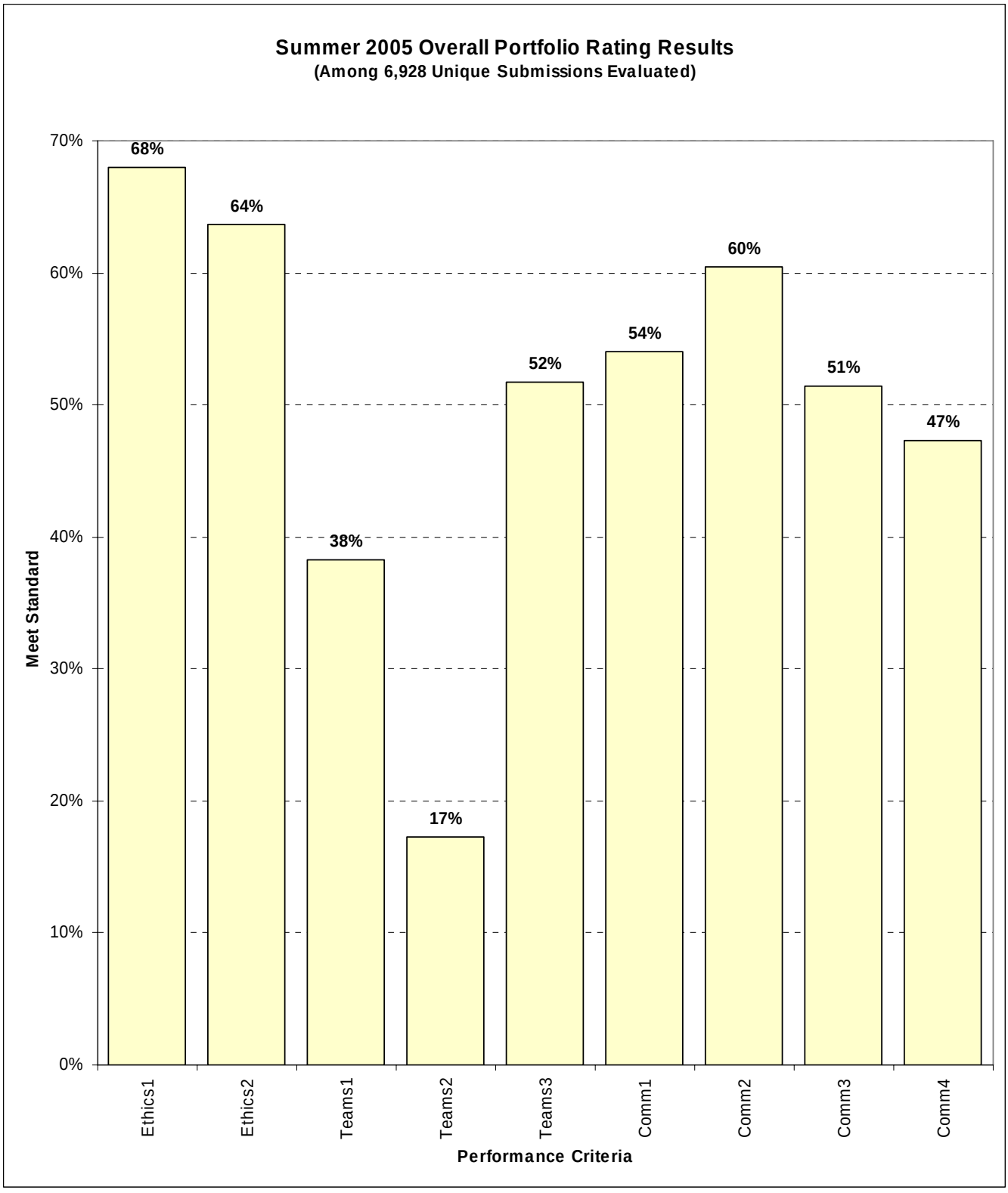
**Rubric**
Description: The student should submit a document/project that has a low percentage of mistakes.

Example: A student may submit a final design report that has been carefully proofread and edited.

Not Acceptable: A level of errors that is distracting or noticeable, such as finding errors in a one or two page sample of text, or in a conspicuous place such as a title, a heading, or an important term.

**Appendix C**

## Summer 2005 Overall Portfolio Rating Results Chart



**Appendix D**

## Summer 2005 Overall Portfolio Rating Results Table

**Overall Summer 2005 Portfolio Rating Results on Ethics, Teams & Communication Learning Objectives:**

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (#) | |
|---|---|---|
| **A. Ethics** | **Yes** | **No** |
| Criterion 1 | 238 | 112 |
| Criterion 2 | 235 | 134 |
| | | |
| **E. Teams** | **Yes** | **No** |
| Criterion 1 | 512 | 827 |
| Criterion 2 | 108 | 518 |
| Criterion 3 | 294 | 274 |
| | | |
| **F. Communication** | **Yes** | **No** |
| Criterion 1 | 410 | 349 |
| Criterion 2 | 493 | 322 |
| Criterion 3 | 331 | 313 |
| Criterion 4 | 690 | 768 |
| **Total** | 3311 | 3617 |

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | |
|---|---|---|
| **A. Ethics** | **Yes** | **No** |
| Criterion 1 | 68% | 32% |
| Criterion 2 | 64% | 36% |
| | | |
| **E. Teams** | **Yes** | **No** |
| Criterion 1 | 38% | 62% |
| Criterion 2 | 17% | 83% |
| Criterion 3 | 52% | 48% |
| | | |
| **F. Communication** | **Yes** | **No** |
| Criterion 1 | 54% | 46% |
| Criterion 2 | 60% | 40% |
| Criterion 3 | 51% | 49% |
| Criterion 4 | 47% | 53% |
| **Total** | 48% | 52% |

**Appendix E**

## Summer 2005 Overall Portfolio Rating Results by Class

### By Class--Overall Summer 2005 Portfolio Rating Results on Ethics, Teams & Communication Learning Objectives:

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (#) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 7 | 7 | 26 | 9 | 55 | 53 | 150 | 43 |
| Criterion 2 | 7 | 3 | 33 | 10 | 70 | 68 | 125 | 53 |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 272 | 443 | 146 | 146 | 43 | 134 | 51 | 104 |
| Criterion 2 | 64 | 174 | 27 | 122 | 4 | 116 | 13 | 106 |
| Criterion 3 | 109 | 91 | 38 | 92 | 70 | 52 | 77 | 39 |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | 86 | 9 | 29 | 251 | 178 | 150 | 56 |
| Criterion 2 | 30 | 119 | 36 | 37 | 298 | 105 | 129 | 61 |
| Criterion 3 | | 30 | 20 | 12 | 249 | 154 | 62 | 117 |
| Criterion 4 | 270 | 373 | 102 | 106 | 211 | 195 | 107 | 94 |
| **Total** | 759 | 1326 | 437 | 563 | 1251 | 1055 | 864 | 673 |

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 50% | 50% | 74% | 26% | 51% | 49% | 78% | 22% |
| Criterion 2 | 70% | 30% | 77% | 23% | 51% | 49% | 70% | 30% |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 38% | 62% | 50% | 50% | 24% | 76% | 33% | 67% |
| Criterion 2 | 27% | 73% | 18% | 82% | 3% | 97% | 11% | 89% |
| Criterion 3 | 55% | 46% | 29% | 71% | 57% | 43% | 66% | 34% |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 0% | 100% | 24% | 76% | 59% | 41% | 73% | 27% |
| Criterion 2 | 20% | 80% | 49% | 51% | 74% | 26% | 68% | 32% |
| Criterion 3 | 0% | 100% | 63% | 38% | 62% | 38% | 35% | 65% |
| Criterion 4 | 42% | 58% | 49% | 51% | 52% | 48% | 53% | 47% |
| **Total** | 36% | 64% | 44% | 56% | 54% | 46% | 56% | 44% |

**Appendix F**

## Overall Comparison Table Summer 2003 vs. Summer 2005

**Overall Summer Portfolio Rating Results Comparison on Ethics, Teams & Communication Learning Objectives:**

**Does the Submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman?**

| R-HIT Learning Objectives | Summer 2003 (n=2,207) | Summer 2005 (n=6,928) | | Meet Standard |
|---|---|---|---|---|
| **A. Ethics** | **Yes** | **Yes** | | **Change in %** |
| Criterion 1 | 33% | 68% | | 35% |
| Criterion 2 | 39% | 64% | | 25% |
| | | | | |
| **E. Teams** | **Yes** | **Yes** | | **No** |
| Criterion 1 | 61% | 38% | | -23% |
| Criterion 2 | 18% | 17% | | 0% |
| Criterion 3 | 83% | 52% | | -31% |
| | | | | |
| **F. Communication** | **Yes** | **Yes** | | **No** |
| Criterion 1 | 18% | 54% | | 36% |
| Criterion 2 | 57% | 60% | | 3% |
| Criterion 3 | 42% | 51% | | 10% |
| Criterion 4 | 33% | 47% | | 15% |
| | | | | |
| **Total** | 41% | 48% | | 7% |

**Appendix G**

## CSSE Department Comparison Table Summer 2003 vs. Summer 2005

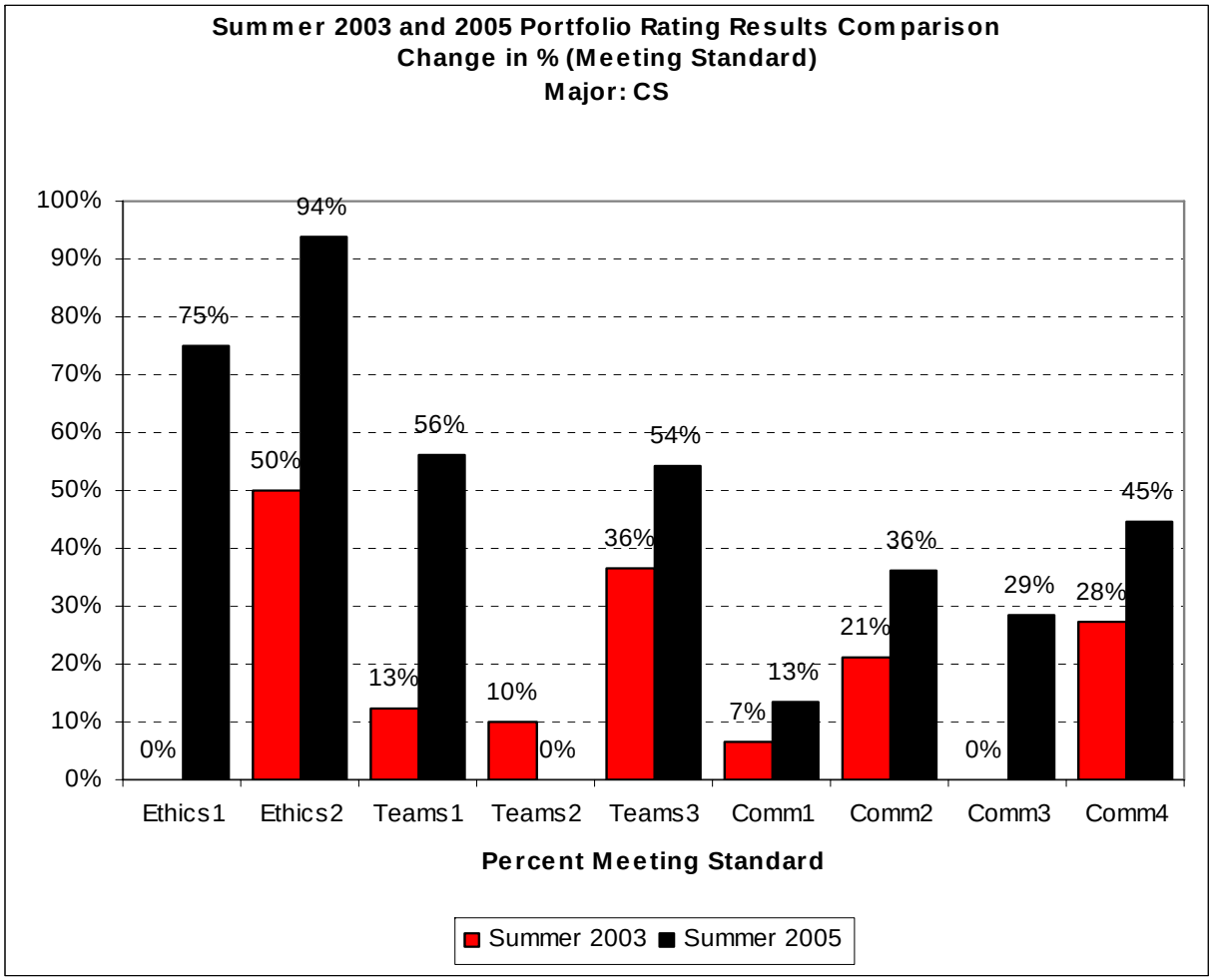
**Portfolio Rating Results For CSSE Department**

**Major: CS**

| R-HIT Learning Objectives | Summer 2003 | CS | | Summer 2005 | CS | | Meet Standard |
|---|---|---|---|---|---|---|---|
| **A. Ethics** | **Yes** | **N** | | **Yes** | **N** | | **Change in %** |
| Criterion 1 | 0% | 3 | | 75% | 8 | | 75% |
| Criterion 2 | 50% | 26 | | 94% | 32 | | 44% |
| | | | | | | | |
| **E. Teams** | **Yes** | **N** | | **Yes** | **N** | | **Change in %** |
| Criterion 1 | 13% | 8 | | 56% | 91 | | 44% |
| Criterion 2 | 10% | 10 | | 0% | 29 | | -10% |
| Criterion 3 | 36% | 11 | | 54% | 35 | | 18% |
| | | | | | | | |
| **F. Communication** | **Yes** | **N** | | **Yes** | **N** | | **Change in %** |
| Criterion 1 | 7% | 15 | | 13% | 15 | | 7% |
| Criterion 2 | 21% | 19 | | 36% | 25 | | 15% |
| Criterion 3 | 0% | 7 | | 29% | 7 | | 29% |
| Criterion 4 | 28% | 40 | | 45% | 85 | | 17% |
| | | | | | | | |
| **Total** | 25% | 139 | | 48% | 327 | | 23% |

**Appendix H**

## CSSE Department Comparison Chart Summer 2003 vs. Summer 2005



**Appendix I**

## Overall Comparison Table by Class Summer 2003 vs. Summer 2005

### Overall Summer Portfolio Rating Results on Ethics, Teams & Communication Learning Objectives:

| R-HIT Learning Objectives | Summer 2003 (n=2,207) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | | | | | | | |
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 0% | 100% | 0% | 100% | 20% | 80% | 42% | 58% |
| Criterion 2 | 0% | 100% | 67% | 33% | 45% | 55% | 40% | 60% |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 71% | 29% | 50% | 50% | 27% | 73% | 57% | 43% |
| Criterion 2 | 24% | 76% | 23% | 77% | 0% | 100% | 8% | 93% |
| Criterion 3 | 91% | 9% | 79% | 21% | 41% | 59% | 81% | 19% |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 2% | 98% | 18% | 82% | 25% | 76% | 5% | 95% |
| Criterion 2 | 29% | 71% | 52% | 48% | 71% | 29% | 51% | 49% |
| Criterion 3 | 0% | 100% | 24% | 76% | 54% | 46% | 10% | 90% |
| Criterion 4 | 31% | 69% | 33% | 68% | 34% | 66% | 39% | 61% |
| **Total** | 40% | 60% | 37% | 63% | 44% | 56% | 39% | 61% |

| R-HIT Learning Objectives | Summer 2005 (n=6,928) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | | | | | | | |
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 50% | 50% | 74% | 26% | 51% | 49% | 78% | 22% |
| Criterion 2 | 70% | 30% | 77% | 23% | 51% | 49% | 70% | 30% |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 38% | 62% | 50% | 50% | 24% | 76% | 33% | 67% |
| Criterion 2 | 27% | 73% | 18% | 82% | 3% | 97% | 11% | 89% |
| Criterion 3 | 55% | 46% | 29% | 71% | 57% | 43% | 66% | 34% |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 0% | 100% | 24% | 76% | 59% | 41% | 73% | 27% |
| Criterion 2 | 20% | 80% | 49% | 51% | 74% | 26% | 68% | 32% |
| Criterion 3 | 0% | 100% | 63% | 38% | 62% | 38% | 35% | 65% |
| Criterion 4 | 42% | 58% | 49% | 51% | 52% | 48% | 53% | 47% |
| **Total** | 36% | 64% | 44% | 56% | 54% | 46% | 56% | 44% |

**Appendix J**

## Summer 2005 Portfolio Rating Results for CSSE Department

### Summer 2005 Portfolio Rating Results For CSSE Department

**Major: CS**

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (#) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | | 3 | | 1 | 1 | 2 | 1 |
| Criterion 2 | | 1 | 8 | | 13 | 1 | 9 | |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 19 | 33 | 21 | 4 | 7 | | 4 | 3 |
| Criterion 2 | | 13 | | | | 6 | | 10 |
| Criterion 3 | 3 | 11 | | | 7 | 1 | 9 | 4 |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | 5 | | 2 | 1 | 2 | 1 | 4 |
| Criterion 2 | 1 | 10 | 2 | 2 | 4 | | 2 | 4 |
| Criterion 3 | | 1 | | 1 | 1 | 1 | 1 | 2 |
| Criterion 4 | 26 | 25 | 6 | 13 | 2 | 4 | 4 | 5 |
| **Total** | 49 | 99 | 40 | 22 | 36 | 16 | 32 | 33 |

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | | 100% | 0% | 50% | 50% | 67% | 33% |
| Criterion 2 | 0% | 100% | 100% | 0% | 93% | 7% | 100% | 0% |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 37% | 63% | 84% | 16% | 100% | 0% | 57% | 43% |
| Criterion 2 | 0% | 100% | | | 0% | 100% | 0% | 100% |
| Criterion 3 | 21% | 79% | | | 88% | 13% | 69% | 31% |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 0% | 100% | 0% | 100% | 33% | 67% | 20% | 80% |
| Criterion 2 | 9% | 91% | 50% | 50% | 100% | 0% | 33% | 67% |
| Criterion 3 | 0% | 100% | 0% | 100% | 50% | 50% | 33% | 67% |
| Criterion 4 | 51% | 49% | 32% | 68% | 33% | 67% | 44% | 56% |
| **Total** | 33% | 67% | 65% | 35% | 69% | 31% | 49% | 51% |

## Summer 2005 Portfolio Rating Results For CSSE Department

### Major: SE

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (#) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | | 1 | | 1 | | 1 | |
| Criterion 2 | | | 1 | | 7 | | 2 | |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 7 | 8 | 5 | 5 | | | | |
| Criterion 2 | 1 | | | | | 1 | | 1 |
| Criterion 3 | 2 | | | | 1 | | 1 | |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | 1 | | 1 | 4 | | 3 | |
| Criterion 2 | | | 2 | 1 | 3 | 1 | 2 | 1 |
| Criterion 3 | | | 1 | | 2 | 2 | | 3 |
| Criterion 4 | 9 | 4 | 5 | 3 | 1 | 3 | 2 | 1 |
| **Total** | 19 | 13 | 15 | 10 | 19 | 7 | 11 | 6 |

| R-HIT Learning Objectives | Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FR | | SO | | JR | | SR | |
| **A. Ethics** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | | | 100% | 0% | 100% | 0% | 100% | 0% |
| Criterion 2 | | | 100% | 0% | 100% | 0% | 100% | 0% |
| | FR | | SO | | JR | | SR | |
| **E. Teams** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 47% | 53% | 50% | 50% | | | | |
| Criterion 2 | 100% | 0% | | | 0% | 100% | 0% | 100% |
| Criterion 3 | 100% | 0% | | | 100% | 0% | 100% | 0% |
| | FR | | SO | | JR | | SR | |
| **F. Communication** | Yes | No | Yes | No | Yes | No | Yes | No |
| Criterion 1 | 0% | 100% | 0% | 100% | 100% | 0% | 100% | 0% |
| Criterion 2 | | | 67% | 33% | 75% | 25% | 67% | 33% |
| Criterion 3 | | | 100% | 0% | 50% | 50% | 0% | 100% |
| Criterion 4 | 69% | 31% | 63% | 38% | 25% | 75% | 67% | 33% |
| **Total** | 59% | 41% | 60% | 40% | 73% | 27% | 65% | 35% |

**Problem Solving Criteria #1: Analyze runtime of code**
**CSSE230, Exam 1 question 1, April 5, 2006.**

**Question and answer:**

**CSSE230 Exam 1 April 5, 2006** **Name:** _____

**Part 1, Closed book, notes, and computer. No network. No communication with others during the exam.** You may do the problems in this exam in any order you want, but must pass in section 1 before using resources (37 pts; suggested time: 25 minutes).

1. (5 points) Find the number of times (big-oh) the **sum++** statement is executed in:

O(__$n^2$__) Part (a) to the right

a)
```
for (i=0; i < n; i++) {
    for (j = 0; j < i; j++) {
        sum++;
    }
}
```

O(_log(n)_) Part (b) to the right

b)
```
for (k = 1; k < n; k = k*2) {
    sum++;
}
```

O(__$n^2$__) (a) and (b) combined.

| | |
|---|---|
| Completely correct | 17 |
| Nested and sequential loops correct. Part not covered by criterion (analyzing code that uses a doubling loop index; part b) incorrect | 4 |
| **Total correct according to criterion** | **21** |
| Nested loops (part a) correct, but sequential loops (part c) incorrect | 2 |
| Nested loops (part a) incorrect, but sequential loops (part c) correct (took max of other runtimes) | 2 |
| Both parts a and c incorrect | 1 |
| **Total incorrect according to criterion** | **5** |
| | |
| **Percentage correct according to criterion** | **80.77** |

# Introductory Physics Lab Assessment 2006

As part of the Core 4 assessment, the Physics Curriculum Committee conducted an assessment of the introductory labs. This report summarizes our procedure and results.

The following four criteria are used for assessing the introductory physics labs:

**1: Use appropriate experimental techniques and procedures**

**2: Keep detailed and accurate records of experimental procedures and data collected**

**3: Estimate experimental uncertainties and use appropriate error propagation techniques**

**4: Draw reasonable conclusions from results**

We conducted the assessment by sampling the RC Circuits reports submitted for the Physics II Lab during the Winter Quarter. Each lab instructor was asked to collect certain student labs which were then copied and returned to the students. The actual assessment was conducted during Spring Quarter by members of the Physics Curriculum Committee.

Three committee members discussed each criteria and came up with guidelines on what would be acceptable and unacceptable student performance. Five reports where evaluated by each assessor and the results discussed. Finally, each assessor evaluated a set of reports independently. Any time there was any question about how a report should be rated, it was discussed as a group.

The results are shown in the table below.

| Criteria | Acceptable | Not Acceptable |
|---|---|---|
| 1 | 28 | 11 |
| 2 | 33 | 6 |
| 3 | 22 | 17 |
| 4 | 15 | 24 |

The 2006 lab assessment results agree fairly well with last year's results. As with the 2005 assessment we are doing well in getting the students to do the labs correctly but not quite as well as in 2005. There has also been a slight decline in getting them to deal with uncertainties correctly. There was noticeable improvement in criteria #2 . Where we still need to improve is in getting the students to draw reasonable conclusions from their results. The biggest problem with the conclusions was the omission of numerical results to support the conclusions. The differences between the two years may be due to assessing two different experiments in two different quarters.

| 2005-2006 Assessment of Design Outcomes | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| **Team** | **Reqts** | **Arch (9)** | **Impl** | **Test** | **Doc** | **Expo** | | |
| **1** | Yes | Yes | Yes | No(1) | Yes | Yes | | |
| **2** | Yes | Yes | Yes | Yes | Yes | Yes | | |
| **3** | Yes | Yes(2) | Yes(2) | No(1) | No(3) | Yes | | |
| **4** | Yes | Yes(4) | Yes(4) | Yes(5) | Yes(6) | Yes | | |
| **5** | Yes | Yes | Yes | Yes! (11) | Yes | Yes | | |
| **6** | No (7) | Yes | Yes | Yes | Yes | Yes | | |
| **7** | Yes | No (10) | Yes | No (10) | No (10) | Yes | | |
| **8** | No (8) | Yes | Yes | No (8) | No (8) | Yes | | |
| **9** | Yes | Yes | Yes | Yes | Yes | Yes | | |
| | | | | | | | | |
| **TOTAL** | 7/9 | 8/9 | 9/9 | 5/9 | 6/9 | 9/9 | | |
| | | | | | | | | |
| **Notes** | | | | | | | | |
| **1** | No test plan or test cases. | | | | | | | |
| **2** | Extensive comments in code describe design. | | | | | | | |
| **3** | But client did not require this. | | | | | | | |
| **4** | Used format specified by client for these. | | | | | | | |
| **5** | Team interacted verbally with client several times a week about test cases. | | | | | | | |
| **6** | Handoff was to client's developers, with documentation in code. | | | | | | | |
| **7** | Did not prepare a written requirements specification. | | | | | | | |
| **8** | Did not turn in evidence of this activity. | | | | | | | |
| **9** | In some cases project plans were used to determine this category. | | | | | | | |
| **10** | The nature of the project (exploratory) did not allow for this. | | | | | | | |
| **11** | Most comprehensive test planning and execution I have ever seen! | | | | | | | |

**CS Exit Interview Responses 2006**

Student: 19 students responding Date: May 19, 2006

1. **What is the best thing that has happened to you as a computer science major?**

- Graduating and getting a successful job (Rockwell Collins)
- Discovered he was a computer scientist (changed majors into CS)
- Got a good job! Really happy with it!
- Being able to do a senior thesis.
- Going to graduate! Learned a lot about computer science.
- Rose-Hulman Ventures – a lot of fun.
- Senior thesis opportunity
- Got a job!
- Breakfast with John Thompson (Schmidt lecturer 2006)
- Got to learn more in depth things about how computers work, languages, SQL. Go to really delve into image processing.
- Getting the ability to apply what I know in the real world.
- PLC.
- Slow realization that it is really cool, that I really like CS, and want to get a Ph.D. someday
- Getting to do some work in development at Ventures.
- All the experience gotten from the projects in the classes.
- Getting interviews – it was easy; having the Rose-Hulman background made it easier.
- Found something I really enjoyed to do.
- I feel like I've become a much more critical thinker.
- Small environment, getting a lot of help from professors, and getting to talk to professors.

2. **What is the worst thing that has happened to you as a computer science major?**

- Experience of a couple of professors. Getting a biomed minor
- Having to repeat Programming Language Concepts.
- Several courses
- Workload
- We don't have an education program
- Claude's classes can be painful
- Weird to come in as a transfer student – felt like the odd man out for awhile (with respect to Rose family feeling)
- Didn't get an internship for the summer after freshman year
- CSSE 120
- Forgotten many things learned in freshman and sophomore years; couldn't remember what "polymorphism" was during a job interview
- Wished there was more emphasis on functional programming

- Don't know
- All the times spent pulling all-nighters.
- Didn't have as much accomplished on thesis as would like to.
- Workload heavily slanted to back end of quarter.
- Frustration from team projects that went badly.
- Having projects where the course didn't teach enough to make the project successful (Databases)
- Sophomore year.
- Nothing.
- A lot of my peers knew more about computer science than I did; felt overwhelmed in some classes.

**3. Please make any comments you think appropriate regarding any faculty member in the department.**

**General comments**

- Impressed with all the faculty I've had. Really good at teaching the courses they did. Really good about meeting outside of class. Good chatting.
- Learned what they were trying to teach in every class.
- Overall, faculty are very good, as people they're excellent. They show passion about what they're doing and care for the students.
- Seem to only get to know faculty really well in junior and senior years.
- Really like the new hires.
- Faculty overall is quite good.
- Like the ability to just drop in and talk.
- Great to call people by their first names.
- All faculty are pretty good.
- Labs being overrun by Legos. If a tour came by it would reflect negatively on the department.

**Individual faculty comments**

*I have removed this section from this version of the summary. If you would like to see the comments seniors made about you in particular, please let me know and I will be happy to send them to you. --Cary*

**4. In your opinion, what are the major strengths in the computer science curriculum?**

- Team based structure
- Consistently further ahead than peers at other schools
- Challenging
- It's broad and yet complete
- Amazing how many different bases we cover, from both required courses and elective courses
- Teaching object oriented in Fundamentals right from the start is a good thing

- CSSE 230, Compilers, Programming Language Concepts, cross-over with math courses, integration of CS/MA double majors, integration with ECE department (ECE 130, 332) really great
- Cover a lot of different topics
- Lot of real world applicability
- Lot of SE exposure is good
- Theory and practical exposure
- Good to get some logic and computer engineering classes.
- SE stuff.
- Topics courses – have taken several.
- Required curriculum goes over the entire span of things; all the electives let you go deep in each area.
- Attention to the students.
- Environment where people can grow and flourish.
- The professors.
- Very diverse, everything from database to AI to image processing.
- Tons of electives you can take to satisfy your interests.
- Really focused on what the best practices are – that's quite good.
- CSSE 490 classes – build courses that adapt with the field is useful.
- Non-rigid, flexible curriculum.
- Introductory courses (120/220/230) are really strong; PLC too. Well structures. Provide good groundwork for whatever you want to do beyond that.
- Project-based courses.
- CSSE 230 is an awesome course.
- Best department as far as faculty interaction and concern – available, easy to get hold of, use first names.
- Flexible, fun projects.
- Electives are really good; a very wide range of topics.
- Focus on math, algorithms really good.
- Ability to get a broad range of computer science; got to learn concepts.

**5. In your opinion, what is the most significant weakness in the department's curriculum?**

- Lack of C. Operating Systems isn't enough.
- Requirements should be taught earlier than junior year.
- When and where particular languages are learned; would have been nice to have learned C before diving into Operating Systems, and Scheme before diving into Programming Language Concepts.
- Should have a Protocols and Formats course, Embedded and I/O Systems course, and a GUI Design course (cross-over with HSS)
- Too career oriented
- Don't have a good grasp of computer science in industry
- None
- Other team members in several courses weren't as strong programmers as I was.

- Some electives are only offered every other year.
- Courses teach you how to learn, but not knowledge.
- Didn't get the curiosity.
- There should be an introductory C course.
- More theoretical underpinnings of things.
- Don't encounter the 490 classes until junior and senior years.
- Wished there were more advanced courses.
- Upper level courses are undirected; can get through courses without picking up anything.
- C needs to be taught better. A lot of people don't understand pointers. Hurts in an interview.
- Writing good, readable code isn't emphasized enough.
- Lack of higher level courses in some terms.
- Coming in with very little experience, felt overwhelmed in early courses.
- Starting with Java seemed problematic. A lot of courses needed C, but no real course that teaches it.
- 371/372 could be done a lot better. Didn't have anything to work with.

6. **What were the most important classes you took?** *(Numbers in parentheses indicate number of students who identified that course as being most important to them.)*

- CSSE 120, Fundamentals of Software Development I (3)
- CSSE 220, Fundamentals of Software Development II (2)
- CSSE 230, Fundamentals of Software Development III (10)
- CSSE 232, Computer Architecture I (3)
- CSSE 304, Programming Language Concepts (12)
- CSSE 332, Operating Systems (5)
- CSSE 333, Database Systems (1)
- CSSE 351, Computer Graphics (2)
- CSSE 371, Software Requirements and Specification (2)
- CSSE 372, Software Project Management (2)
- CSSE 404, Compiler Construction (4)
- CSSE 413, Artificial Intelligence (3)
- CSSE 432, Computer Networks (1)
- CSSE 473, Design and Analysis of Algorithms (3)
- CSSE 474, Theory of Computation (2)
- CSSE 490, Swarm Intelligence (1)
- CSSE 497-498-499, Senior Project I-II-III (1)
- ECE 332, Computer Architecture II (1)
- Courses with large projects (1)

7. **What classes did you enjoy the most?** *(Numbers in parentheses indicate number of students who identified that course as being enjoyable to them.)*

- CSSE 120, Fundamentals of Software Development I (2)

- CSSE 230, Fundamentals of Software Development III (1)
- CSSE 232, Computer Architecture I (1)
- CSSE 304, Programming Language Concepts (8)
- CSSE 325, Fractals and Chaotic Dynamical Systems (1)
- CSSE 332, Operating Systems (1)
- CSSE 333, Database Systems (2)
- CSSE 351, Computer Graphics (9)
- CSSE 371, Software Requirements and Specification (1)
- CSSE 404, Compiler Construction (3)
- CSSE 413, Artificial Intelligence (7)
- CSSE 432, Computer Networks (3)
- CSSE 442, Computer Security (2)
- CSSE 451, Advanced Computer Graphics (2)
- CSSE 473, Design and Analysis of Algorithms (1)
- CSSE 474, Theory of Computation (1)
- CSSE 479, Cryptography (1)
- CSSE 490, Great Papers in Computer Science and Software Engineering (1)
- CSSE 490, Swarm Intelligence (3)
- CSSE 497-498-499, Senior Project I-II-III (1)
- Japanese (1)
- Topics in Discrete Mathematics (1)

8. **What were your least favorite classes?** *(Numbers in parentheses indicate number of students who identified that course as being least favorite to them.)*

- CSSE 120, Fundamentals of Software Development I (1)
- CSSE 220, Fundamentals of Software Development II (3)
- CSSE 230, Fundamentals of Software Development III (3)
- CSSE 232, Computer Architecture I (2)
- CSSE 304, Programming Language Concepts (5)
- CSSE 332, Operating Systems (1)
- CSSE 333, Database Systems (5)
- CSSE 371, Software Requirements and Specification (4)
- CSSE 372, Software Project Management (4)
- CSSE 413, Artificial Intelligence (1)
- CSSE 432, Computer Networks (1)
- CSSE 442, Computer Security (1)
- CSSE 474, Theory of Computation (2)
- CSSE 481, Web-Based Information Systems (1)
- CSSE 497-498-499, Senior Project I-II-III (1)
- ECE 130, Introduction to Logic Design (1)
- ES 202, Fluid and Thermal Systems (1)
- MA 111-112, Calculus I-II (1)
- Psychology with Dr. Brophy (1)

**9. What is the most important change needed in the department?**

- Teaching more of the basics earlier on
- Not dumping students into everything
- Programming Language Concepts course – trying to stuff too much into one course, and there's more to cover as well; split into two courses
- Need some new faculty (swap some out)
- Missing a lot of the current, really useful stuff
- More opportunities for students to be involved with faculty research
- Can't take all the courses in the department! There's a lot I would have liked to take.
- Get website up to date and keep it up to date
- Not sure
- Told to go easy on grading when students should be challenged more
- Did it already – CSSE 221
- More practical electives (e.g. web technology, PSP, Python)
- Introductory C course
- .NET integration into curriculum
- Getting a solid curriculum is needed; a lot of restructuring going on while I was here
- A better application of SE; 370 series doesn't express it in an optimal way.
- None.
- Don't know.
- Quality of teaching could be brought up a level in most courses.
- Getting a good core of faculty to stay awhile.
- Can't think of any.
- None. Everything's fine, nice and open. The open environment is good.
- Changing 371/372.
- Getting good projects for seniors.
- Didn't do enough individual work.

**10. Rate how well you feel you have satisfied each of the following CS program outcomes on a scale of 1 (low) to 5 (high):** *(Numbers following the colons indicate the number of students who gave that numerical ranking for how well they feel they have satisfied that outcome.)*

Apply effectively a variety of programming languages, programming paradigms, operating systems, networks, and software development tools.

1: 0 2: 0 3: 5 4: 6 5: 8 Average: 4.16

Anticipate complexities and problems involved in the development of large software systems.

1: 0 2: 0 3: 3 4: 14 5: 2 Average: 3.95

Analyze requirements, design software that satisfies those requirements, and implement that software.

1: 0 2: 0 3: 1 4: 9 5: 9 Average: 4.42

Analyze problems using ideas of problem complexity, models of computation, and decidability.

1: 1 2: 2 3: 6 4: 7 5: 3 Average: 3.47

Design algorithms using a variety of paradigms

1: 0 2: 2 3: 5 4: 8 5: 4 Average: 3.74

Analyze algorithms in terms of correctness, as well as time and space efficiency.

1: 1 2: 1 3: 6 4: 6 5: 5 Average: 3.68

Communicate effectively, both verbally and in writing.

1: 0 2: 0 3: 3 4: 10 5: 6 Average: 4.16

Evaluate and discuss the legal, social, and ethical aspects of significant events that arise in the computing industry.

1: 0 2: 3 3: 7 4: 8 5: 1 Average: 3.37

Collaborate effectively in small teams.

1: 0 2: 0 3: 1 4: 2 5: 16 Average: 4.79

**11. Did you do a senior thesis or project?**

- Project: 15
- Thesis: 4

**12. What are your post graduation plans?** *(Numbers in parentheses indicate how many students are going to that particular location, if more than 1.)*

Project students:

- Rockwell Collins (2)
- Still looking (2)
- Green Hills Software
- Northwestern University – MS in Education
- Dairyland Healthcare Solutions (Glenwood, MN)
- Beckman Coulter
- Carnegie Mellon University – MS in HCI

- Software Engineering Professionals
- University of New Hampshire – Ph.D. in CS
- Microsoft (2)
- Mantech International (government contractor for NSA in Maryland)
- The Matrixx Group (Evansville, IN)

Thesis students:
- Microsoft
- Indiana University – Ph.D. in CS
- U.S. Air Force
- Carlson Software (Boston)

**13. Any closing comments?**

- CS department has done an excellent job preparing me. Working at Rockwell Collins I am better than other people there because I've learned in a different way (they're older).
- Wish he knew he was a computer scientist when he came here (started out as chemical engineering, changed to CS beginning of junior year). Liked ease of transition to CS even though it meant another year at Rose.
- Department labs – don't like them.
- Things aren't accessible.
- Love OS/Security lab – but fall quarter no classes in lab, always locked, couldn't get in to use the equipment.
- FR/SO years – would be nice to have log in to CSSE network – can it be automatic for all CSSE students?
- Would like to have access to workstations that don't normally have access to – new Mac, Sun that works correctly (it's own OS, GUI).
- Mike McLeish – either overworked or underpaid.
- 371 & 372 should be combined – not enough material for two courses.
- PLC should not be required.
- Never learned what NP-complete and NP-hard are – profs would use the terms but never explain them.
- Senior projects are grade inflated – client complained a lot and they still got an A.
- 333 should teach how to make RDBMS, not just use it.
- Return homework – CS profs not good about this.
- Should be daily or weekly homework in most courses.
- More emphasis on code optimization and analysis.
- It was fun. Made the right choice not going to Carnegie Mellon.
- Overall the experience was great.
- Would choose Rose-Hulman again.
- Professors are really good.
- Had Databases last time Dale taught it.
- Use less PowerPoint in class.
- Use more anonymous feedback throughout courses.

- Need a course on code construction.
- More CS events in department (e.g. programming competitions, tech talks by students and faculty).
- Can the wiki be made so students can update things, e.g. appropriate web pages?
- Online discussion forum for the department.
- Need fast servers.
- Need alternative OS machines in labs – improved documentation for what's in the labs.
- Source control needs to be better advertised and documented.
- Wikis and bug tracking.
- Really enjoyed it. Think the faculty here is great, every one of them.
- Pretty satisfied with experience here. Very educational and pleasant.
- Senior project turned out to be too easy; unchallenging, dragged out over entire year.
- Tell people CSSE 230 is important.

# IRPA Alumni Survey Results (part 3)

**Results**

Prepared for:

Cary Laxer
Department Head
Computer Science and Software Engineering

Prepared by:

Shannon Sexton, Julia Williams, & Timothy Chow
Office of Institutional Research, Planning and Assessment
Rose-Hulman Institute of Technology

March 2, 2006

**ROSE-HULMAN INSTITUTE OF TECHNOLOGY**
**OFFICE OF INSTITUTIONAL RESEARCH, PLANNING AND ASSESSMENT**
5500 WABASH AVENUE-CM11 TERRE HAUTE, INDIANA 47803-3999
TELEPHONE: (812) 877-8816 FAX: (812) 877-8931 EMAIL: IRPA@rose-hulman.edu

**IRPA** **Alumni Survey Results** (part 3)

# Table of Contents


PAGE

**METHODOLOGY**

Survey Design Process ..... 1
Statistical Analysis ..... 1
Report Format ..... 1

**FINDINGS**

Personal Demographic Information ..... 2
Educational Background ..... 6
Employment ..... 9
Program and Departmental Specific Items ..... 11

**APPENDIXES**

A. Graduate Institutions Attended by CS Alumni
B. Graduate Degree Specializations of CS Alumni
C. Employment History of CS Alumni

# Methodology

## Survey Design Process

In August 2005, the dean and department heads of Rose-Hulman Institute of Technology agreed to develop the first Institute-wide Alumni Survey. The purpose of the survey was to collect information from alumni for use in various reports, including the ABET Self Study Reports (for program accreditation) and the North Central Association/AQIP Systems Portfolio (for institutional accreditation).

The survey instrument was developed with assistance from the staff of the Office of Institutional Research, Planning, and Assessment. The IRPA staff first examined standard alumni survey instruments to determine potential question formats and types. A draft of the Institutional Survey was then submitted to department heads and the dean for input. The Institutional Survey included questions that were common among all departments, including questions about demographics, current employment, continuing education, and Institutional learning outcomes. Some departments, however, submitted program-specific questions; these questions were related to Program Educational Objectives and Program Educational Outcomes relevant to ABET accreditation. The following departments provided department-specific items for the survey: Applied Biology/ Biomedical Engineering; Applied Optics/ Optical Engineering; Chemical Engineering; Civil Engineering; Computer Science/Software Engineering; and Mathematics. All other departments used the Institutional version of the survey without additional items added.

The survey was published online by IRPA staff on October 14, 2005. In order to invite participation, department heads sent email and/or a letter to their program's alumni asking them to complete the survey. Email addresses were obtained through the RHIT Alumni Office. Some departments included a reference to the survey, as well as the URL, on their departmental homepages or newsletter. Given the large number of alumni who have graduated from some programs, IRPA encouraged the use of a statistical sample of alums, rather than asking all alumni to participate. In additional to the statistical sampling, alumni who were 5, 10, 15, and 20 years from graduation were targeted. This again was in an effort to narrow the pool of respondents while still ensuring quality data collection[1]. The survey remained open until December 15, 2005.

## Statistical Analysis

Frequency data and descriptive statistics were used to calculate the findings of the survey. As such, all data is presented in terms of percentages and total numbers of students.

## Report Format

This report is organized by the order of questions that comprised the survey. For each question, responses at the institutional level are reported first. Program responses are reported second. Each department will see data broken down by program for their programs only.

[1] This process yielded an overall response rate of 23%. The CS response rate was 25%.

## Personal Demographic Information

On the "Background Information" section of the survey alumni were asked to report general demographic information such as gender and ethnicity, RHIT educational experience including major and minor, and post RHIT educational experiences such as additional degrees completed and certificates earned (see Table 1 for a breakdown of gender and ethnicity)[2].

**Table 1**
**Personal Demographic Information**

| Demographic | * Number of Respondents | Percentage of Total Respondents |
|---|---|---|
| **Gender** | | |
| | | |
| Male | 875 | 89% |
| Female | 106 | 11% |
| | | |
| **Race/Ethnicity** | | |
| | | |
| African American | 8 | 1% |
| Asian | 20 | 2% |
| Caucasian | 890 | 91% |
| Hispanic | 9 | 1% |
| Native American | 2 | > 1% |
| Pacific Islander | 2 | > 1% |
| Other | 15 | 2% |

* Institute N = 979

In terms of RHIT educational experience, the majority of alumni responding to the survey were Chemical Engineering majors followed by Electrical Engineering, Mathematics, Mechanical Engineering, and Computer Science. Of alumni reporting a minor while at RHIT, the majority reported a minor in Economics. This was followed by minors in German and Mathematics.

Looking at CS alumni only, the majority reported a minor in Economics followed by Political Science (see Tables 2, 3a & 3b for breakdown of majors and minors).

[2] While alumni were asked to report their current state of residence, due to a coding error we do not have that data to report.

**Table 2**
**Respondents by Major**

| Major | * Number of Respondents | Percentage of Total Respondents |
|---|---|---|
| **Applied Biology and Biomedical Engineering** | | |
| | | |
| Applied Biology | 6 | 1% |
| Biomedical Engineering | 9 | 1% |
| | | |
| **Air Force ROTC** | 3 | > 1% |
| | | |
| **Chemical Engineering** | 299 | 31% |
| | | |
| **Chemistry** | | |
| | | |
| Biochemistry and Molecular Biology | 5 | > 1% |
| Chemistry | 69 | 7% |
| | | |
| **Civil Engineering** | | |
| | | |
| Civil Engineering | 44 | 4% |
| Environmental Engineering | 7 | 1% |
| | | |
| **Computer Science and Software Engineering** | | |
| | | |
| Computer Science | 130 | 13% |
| Software Engineering | 2 | > 1% |
| | | |
| **Electrical and Computer Engineering** | | |
| | | |
| Computer Engineering | 33 | 3% |
| Electrical Engineering | 165 | 17% |
| | | |
| **Humanities and Social Sciences** | | |
| | | |
| Economics | 26 | 3% |
| | | |
| **Mathematics** | 136 | 14% |
| | | |
| **Mechanical Engineering** | 130 | 13% |
| | | |
| **Physics and Optical Engineering** | | |
| | | |
| Applied Optics | 22 | 2% |
| Optical Engineering | 4 | > 1% |
| Physics | 38 | 4% |

* Institute N = 979
Note: Both single and double majors are accounted for.

**Table 3a**
**Technical Minors[3]**

| Minor | * Number of RHIT Respondents | Percentage of Total RHIT Respondents | * Number of CSSE Respondents | Percentage of Total CSSE Respondents |
|---|---|---|---|---|
| **Applied Biology and Biomedical Engineering** | | | | |
| | | | | |
| Applied Biology | 13 | 1% | | |
| Biomedical Engineering | 18 | 2% | | |
| | | | | |
| **Chemical Engineering** | | | | |
| | | | | |
| Biochemical Engineering | 6 | 1% | | |
| Chemical Engineering | 7 | 1% | | |
| | | | | |
| **Chemistry** | | | | |
| | | | | |
| Biochemistry and Molecular Biology | 4 | > 1% | | |
| Chemistry | 7 | 1% | | |
| | | | | |
| **Civil Engineering** | | | | |
| | | | | |
| Environmental Engineering | 23 | 2% | | |
| | | | | |
| **Computer Science and Software Engineering** | | | | |
| | | | | |
| Computer Science | 27 | 3% | 3 | 3% |
| Software Engineering | 1 | > 1% | | |
| | | | | |
| **Electrical and Computer Engineering** | | | | |
| | | | | |
| Computer Engineering | 2 | > 1% | | |
| Electrical Engineering | 13 | 1% | 1 | 1% |
| | | | | |
| **Mathematics** | 33 | 3% | 2 | 2% |
| | | | | |
| **Physics and Optical Engineering** | | | | |
| | | | | |
| Applied Optics | 11 | 1% | | |
| Physics | 15 | 2% | | |

* Institute N = 979; CS N = 111

[3] Cells that are blank in the tables of this report did not have an alumnus indicate that response.

**Table 3b**
**Humanities and Social Sciences Minors**

| Minor | * Number of RHIT Respondents | Percentage of Total RHIT Respondents | * Number of CS Respondents | Percentage of Total CS Respondents |
|---|---|---|---|---|
| | | | | |
| Anthropology | 2 | > 1% | | |
| East Asian Studies | 2 | > 1% | | |
| Economics | 73 | 7% | 9 | 8% |
| European Studies | 1 | > 1% | | |
| German | 57 | 6% | 5 | 5% |
| History | 14 | 1% | 2 | 2% |
| Japanese | 14 | 1% | 6 | 5% |
| Language & Literature | 23 | 2% | 4 | 4% |
| Latin American Studies | 3 | > 1% | 1 | 1% |
| Philosophy & Religion | 13 | 1% | 3 | 3% |
| Political Science | 28 | 3% | 7 | 6% |
| Psychology | 10 | 1% | 1 | 1% |
| Science, Technology & Society | 2 | > 1% | 1 | 1% |
| Spanish | 24 | 2% | 1 | 1% |
| Other | 31 | 3% | | |

* Institute N = 979; CS N = 111

## Educational Background

Regardless of the major, the majority of alumni responding to the survey graduated in the last 5 years. This pattern was the same looking at only CS alumni response rate. Fifty-two percent of RHIT alumni responding to the survey reported earning some type of graduate degree. Twenty-five percent of ME alumni reported earning a graduate degree with the majority earning a MS.

Of CS alumni who completed a graduate degree, the majority did so in Computer Science (n = 14), Engineering Management (n = 2), or Marketing (n = 2). These alumni attended Indiana University (n = 7) and University of Dallas (n = 2) most often. (For a complete listing of graduate degree institutions and specializations for CS alumni refer to Appendixes A & B.) The majority of RHIT alumni completing a graduate degree did so within 5 years of graduating from Rose-Hulman. The same is true looking at only CS alumni.

Looking at certificates and special programs alumni completed while at Rose-Hulman, most CS alumni completed a Technical Translator Certificate program (n = 5) followed by an Imaging Systems Certificate (n = 3). The most commonly reported professional certificate earned by RHIT alumni was an Engineering Intern certificate. CS alumni most often reported earning a professional certification in Six Sigma. Thirty-three percent of RHIT alumni reported taking additional courses past their undergraduate degree not counting toward a graduate program. Thirty-eight percent of CS alumni reported taking additional courses not counting toward a graduate program. Ten percent of RHIT alumni and 4% of CS alumni reported completing a workshop not counting toward a degree (see Table 4 for educational background breakdown).

**Table 4**
**Educational Background**

| Educational Item | * Number of RHIT Respondents | Percentage of Total RHIT Respondents | * Number of CS Respondents | Percentage of Total CS Respondents |
|---|---|---|---|---|
| **RHIT Graduation Year** | | | | |
| | | | | |
| Prior to 1950 | 7 | 1% | | |
| 1950s | 38 | 4% | | |
| 1960s | 80 | 8% | | |
| 1970s | 101 | 10% | 5 | 5% |
| 1980s | 198 | 20% | 19 | 17% |
| 1990s | 251 | 26% | 34 | 31% |
| 2000s | 285 | 29% | 49 | 44% |
| | | | | |
| **Graduate Degree** | | | | |
| | | | | |
| MA | 25 | 3% | 2 | 2% |
| MS | 224 | 23% | 16 | 14% |
| MBA | 110 | 11% | 8 | 7% |
| MEng | 10 | 1% | | |
| PhD | 75 | 8% | 2 | 2% |
| MD | 9 | 1% | | |
| JD | 17 | 2% | | |
| Other | 36 | 4% | | |
| | | | | |
| **** Time to Complete Graduate Degree** | | | | |
| | | | | |
| 1-5 | 201 | 40% | 9 | 32% |
| 6-10 | 120 | 24% | 8 | 28% |
| 11-15 | 41 | 8% | 3 | 11% |
| 16 or more | 36 | 7% | 3 | 11% |
| | | | | |
| **Professional Certificates Earned** | | | | |
| | | | | |
| Actuarial Fellow | 6 | 1% | | |
| Engineering Intern | 160 | 16% | | |
| Professional Engineer | 101 | 10% | | |
| Secondary Education Certification | 20 | 2% | 1 | 1% |
| Six Sigma Certification | 94 | 10% | 6 | 5% |
| | | | | |
| **Certificate(s) / Special Program(s) at R-HIT** | | | | |
| | | | | |
| Fast Track Calculus | | | 2 | 2% |
| Imaging Systems Certificate | | | 3 | 3% |
| Management Studies Program | | | 1 | 1% |
| Technical Translator's Certificate Program (German) | | | 3 | 3% |
| Technical Translator's Certificate Program (Russian) | | | 2 | 2% |
| | | | | |
| **Additional Courses Completed** | | | | |
| | | | | |
| None | 520 | 53% | 51 | 46% |
| 1-4 | 187 | 19% | 13 | 12% |
| 5-10 | 132 | 14% | 29 | 26% |
| | | | | |

| Additional Workshops Completed | | | | |
|---|---|---|---|---|
| | | | | |
| None | 739 | 76% | 84 | 76% |
| 1-4 | 74 | 8% | 4 | 4% |
| 5-10 | 19 | 2% | | |
| | | | | |

* Institute N = 979; CS N = 111

** Percentage based on number of alumni obtaining a graduate degree (Institute n = 506; CS n = 28)

## Employment

Overall, RHIT alumni reported working in manufacturing/production most often. CS alumni reported working in software development most often. Other high frequency areas included software development and consulting for RHIT alumni and consulting for CS alumni. Alumni were able to check more than one industry area resulting in a larger number of responses than alumni.

The majority of alumni, both RHIT alumni and CS alumni specifically, reported their current position is in their major field. They work most in large organizations with more than 10,000 employees and earn between $60,000-$79,999 per year (see Table 5 for a breakdown of employment area).

**Table 5**
**Employment Demographics**

| Employment Item | * Number of RHIT Respondents | Percentage of Total RHIT Respondents | * Number of CS Respondents | Percentage of Total CS Respondents |
|---|---|---|---|---|
| **Industry Area** | | | | |
| | | | | |
| Actuary | 8 | 1% | | |
| Construction | 26 | 3% | | |
| Consulting | 103 | 11% | 10 | 9% |
| Customer Service/Support | 7 | 1% | | |
| Development | 73 | 7% | 5 | 5% |
| Education | 54 | 6% | 7 | 6% |
| Finance | 26 | 3% | 7 | 6% |
| Government | 37 | 4% | 5 | 5% |
| Homemaker | 2 | > 1% | | |
| Law | 20 | 2% | | |
| Management | 77 | 8% | 2 | 2% |
| Manufacturing/Production | 166 | 17% | 2 | 2% |
| Marketing/Sales | 46 | 5% | 3 | 3% |
| Medicine | 33 | 3% | 4 | 4% |
| Military | 29 | 3% | 5 | 5% |
| Network/Systems Support | 17 | 2% | 8 | 7% |
| Operations/Maintenance | 29 | 3% | 2 | 2% |
| Product Support | 18 | 2% | 1 | 1% |
| Product Design | 92 | 9% | 4 | 4% |
| Project Management | 69 | 7% | 5 | 5% |
| Research | 97 | 10% | 8 | 7% |
| Statistician | 10 | 1% | | |
| Software Development | 135 | 14% | 65 | 59% |
| Testing/Quality Assurance | 42 | 4% | 3 | 3% |
| Other | 102 | 10% | | |
| | | | | |
| **Current Position** | | | | |
| | | | | |
| Within my major field | 727 | 74% | 97 | 87% |
| Within another related major field | 28 | 3% | | |
| Outside my major field | 111 | 11% | 7 | 6% |
| Retired | 59 | 6% | | |
| I am not currently employed | 54 | 6% | 7 | 6% |
| | | | | |

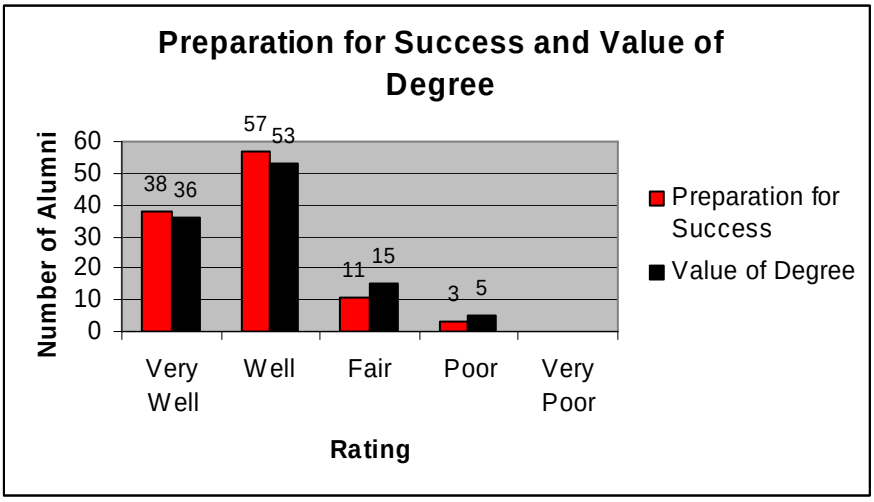
| **Size of Employing Organization** | | | | |
|---|---|---|---|---|
| | | | | |
| 10 or less | 27 | 3% | 5 | 5% |
| 11-50 | 59 | 6% | 5 | 5% |
| 51-150 | 35 | 4% | 7 | 6% |
| 151-500 | 113 | 12% | 14 | 13% |
| 501-1,000 | 79 | 8% | 10 | 9% |
| 1,001-2,500 | 98 | 10.0% | 13 | 12% |
| 2,501-5,000 | 85 | 9% | 11 | 10% |
| 5,001-10,000 | 64 | 7% | 6 | 5% |
| Over 10,000 | 284 | 29% | 30 | 27% |
| Not Applicable | 124 | 13% | 8 | 7% |
| | | | | |
| **Current Annual Salary** | | | | |
| | | | | |
| Below $40,000 | 6 | 1% | 2 | 2% |
| $40,000-$59,999 | 87 | 9% | 19 | 17% |
| $60,000-$79,999 | 424 | 43% | 68 | 61% |
| $80,000-$99,999 | 233 | 24% | 13 | 12% |
| $100,000-$119,999 | 59 | 6% | | |
| $120,000-$139,999 | 31 | 3% | | |
| $140,000-$159,999 | 2 | > 1% | | |
| $160,000 or more | 6 | 1% | 1 | 1% |
| Not employed full-time | 73 | 8% | 6 | 5% |
| I prefer not to respond | 7 | 1% | 1 | 1% |

* Institute N = 979; CS N = 111

## Program and Departmental Specific Items

CS alumni mostly reported their undergraduate degree prepared them for success in their current position well. They also rated the overall value of their degree (when comparing cost of the degree to its contribution to alumni success) as "well" most often (see Figure 1).

**Figure 1**
**Preparation for Success**




CS alumni were very likely to recommend their undergraduate program to others and were satisfied overall with their undergraduate education (see Figures 2 & 3).

**Figure 2**
**Likelihood of Recommending Undergraduate Program to Others**

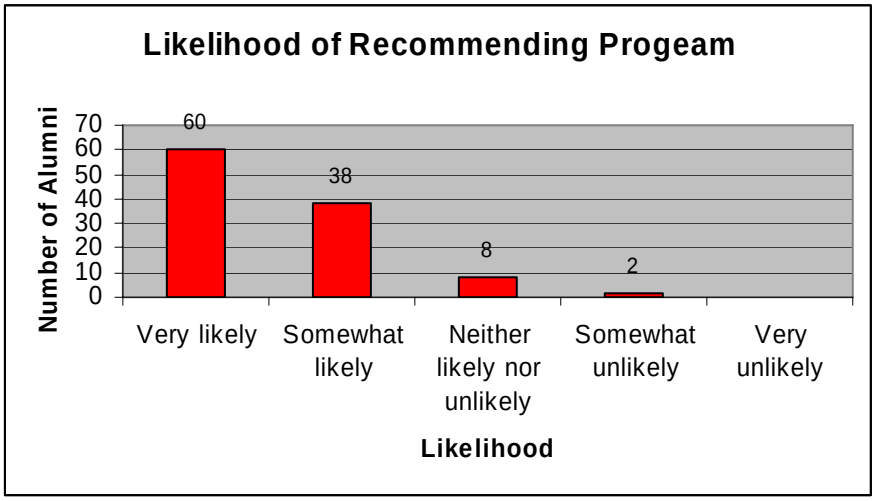
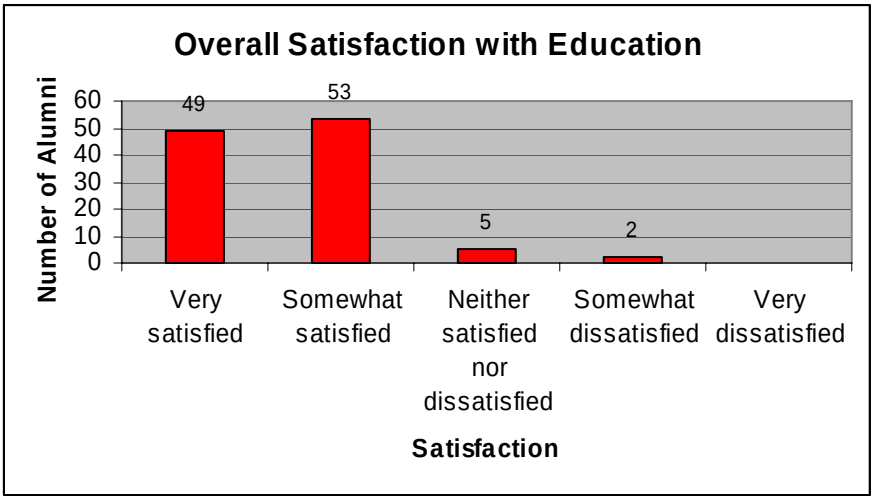


**Figure 3**
**Overall Satisfaction with RHIT Education**



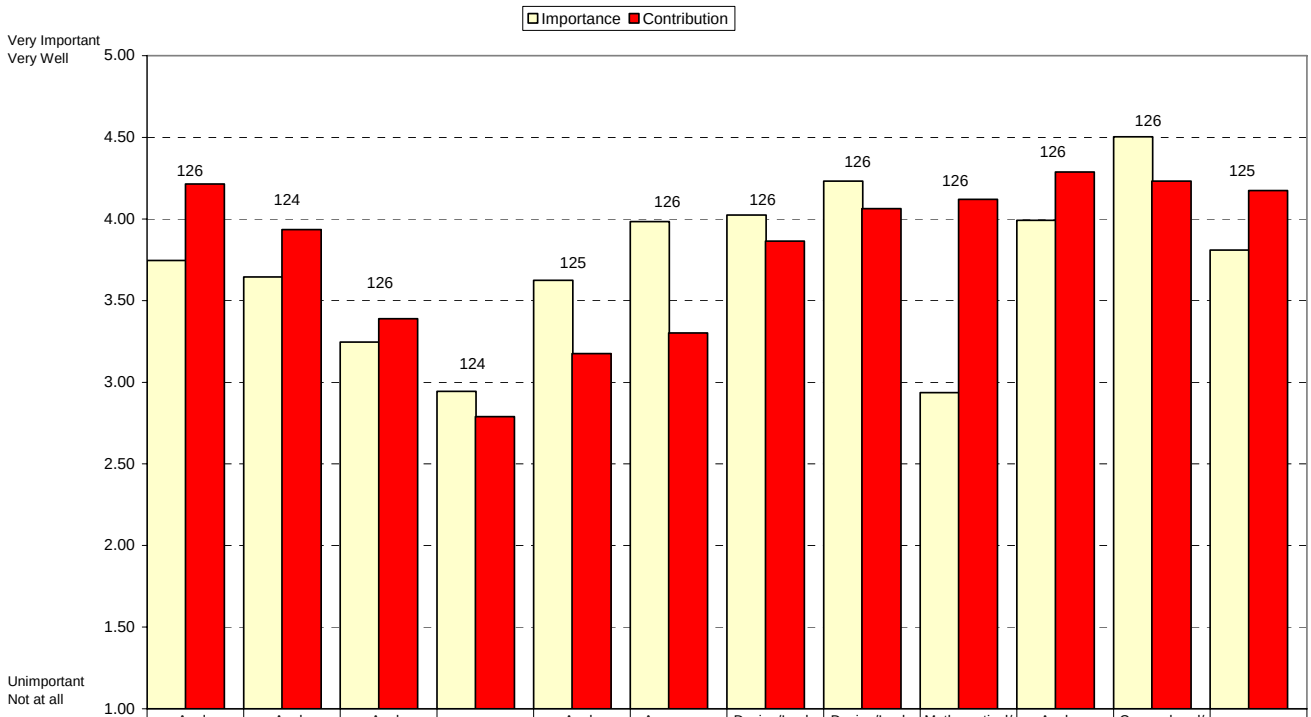

In addition to the Institute-wide learning outcomes, CS alumni were asked to rate the importance and the contribution of RHIT to CS specific learning outcomes. CS alumni were asked to rate their:

1. depth of knowledge to apply a variety of programming languages
2. depth of knowledge to apply a variety of programming paradigms
3. depth of knowledge to apply a variety of operating systems
4. depth of knowledge to apply a variety of networks
5. depth of knowledge to apply a variety of software development tools
6. awareness of the complexities and problems involved in the development of large software systems
7. ability to design and implement software tools to solve problems
8. ability to design and implement software solutions to solve problems
9. familiarity with the mathematical and theoretical underpinnings of computing
10. ability to analyze both problems and algorithms
11. ability to comprehend, evaluate, and learn new technologies
12. appropriate depth of knowledge to enable learning of new areas of computer science

They rated the contribution of RHIT of each of these as “barely” to “well.” They rated the importance of each of these to their current position as “moderate” to “important” (see Figure 4 for a pictorial representation of alumni ratings).

## Figure 4

**Average Ratings on Importance and R-HIT Contribution (Major = CS)**
**(Departmental Learning Outcomes)**




| | Apply Programming Languages | Apply Programming Paradigms | Apply Operating Systems | Apply Networks | Apply Development Tools | Awareness Complexities/ Problems | Design/Implement Software Tools | Design/Implement Software Solutions | Mathematical/ Theoretical Underpinnings | Analyze Problems/Algorithms | Comprehend/ Evaluate/Learn New Tech | Depth of Knowledge |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Importance | 3.75 | 3.65 | 3.25 | 2.94 | 3.62 | 3.98 | 4.02 | 4.23 | 2.94 | 3.99 | 4.50 | 3.81 |
| Std. Deviation | 1.24 | 1.14 | 1.24 | 1.18 | 1.14 | 1.23 | 1.07 | 1.05 | 1.08 | 1.00 | 0.76 | 1.12 |
| Contribution | 4.21 | 3.94 | 3.39 | 2.79 | 3.18 | 3.30 | 3.87 | 4.06 | 4.12 | 4.29 | 4.23 | 4.17 |
| Std. Deviation | 0.88 | 0.98 | 1.01 | 1.02 | 1.04 | 1.15 | 1.02 | 0.92 | 0.87 | 0.78 | 0.87 | 0.85 |

CS alumni were also asked to rate the amount of various types of courses in the CSSE curriculum, how well the CSSE department stays in contact with alumni, the importance of staying in contact with alumni, and beneficial services and opportunities for CSSE alumni. (Beneficial services and opportunities can be found in Appendix D and E.) CS alumni reported thinking the CSSE department should require the same amount of theory, more practice, more project work, the same amount of mathematics, the same amount of humanities, the same number of CSSE electives, and the same number of general electives (See Figure 5 for a breakdown of course requirements). The majority of CS alumni reported the CSSE department is doing well keeping in contact with alumni and that it is important to do so (See Figure 6 for a breakdown of alumni contact ratings).

**Figure 5**
**Amount of Courses Required by the CSSE Department**

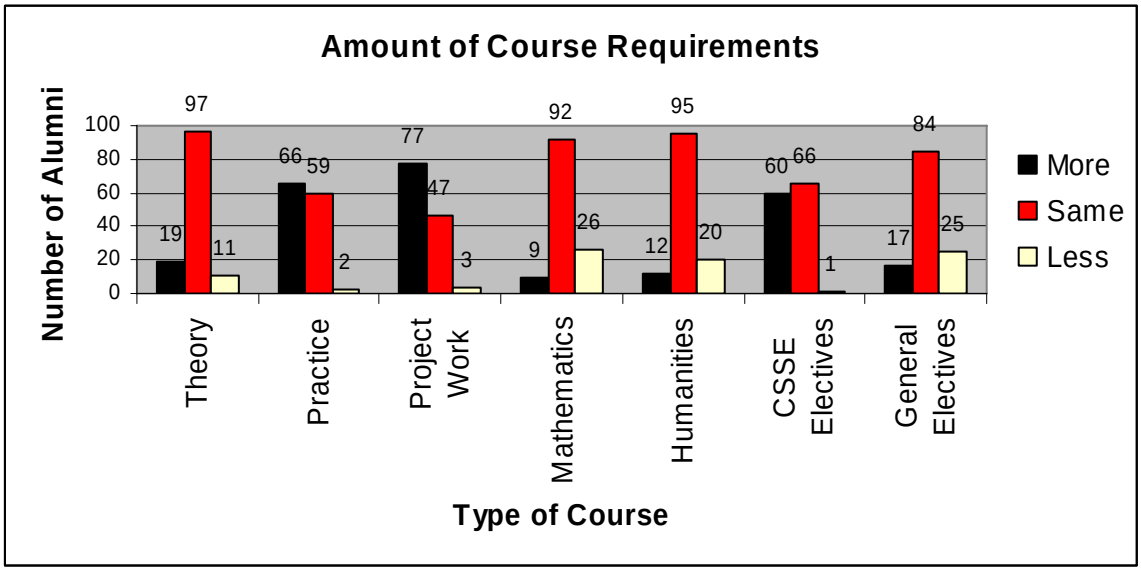
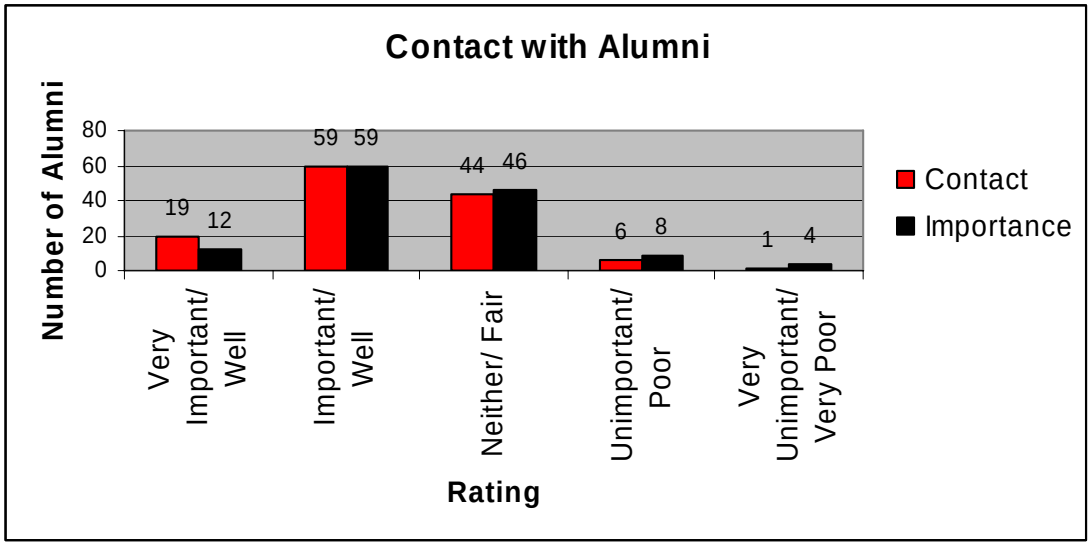



**Figure 6**
**Amount and Importance of Contact with Alumni by the CSSE Department**



## Appendix A

### Graduate Institutions Attended by CS Alumni

| * Number of Alumni | Graduate Institution |
|---|---|
| | Ball State University |
| | Butler University |
| | Columbia University |
| | Florida Atlantic University |
| | Illinois Institute of Technology |
| 7 | Indiana University |
| | Kennesaw State University |
| | National Technology Institute |
| | Rose-Hulman Institute of Technology |
| | Tufts University |
| | University of Charleston |
| 2 | University of Dallas |
| | University of Illinois |
| | University of Massachusetts |
| | University of Michigan |
| | University of Missouri-Rolla |
| | University of Notre Dame |
| | University of Tennessee |
| | University of Utah |

* Note: Institutions without a numerical distinction had only one alumnus attend.

## Appendix B

### Graduate Degree Specializations of CS Alumni

| * Number of Alumni | Graduate Degree Specialization |
|---|---|
| | Computational Linguistics |
| 14 | Computer Science |
| 2 | Engineering Management |
| | Finance |
| | Management |
| 2 | Marketing |
| | Programming Languages & Computer Security |
| | Software Engineering |

* Note: Institutions without a numerical distinction had only one alumnus attend.

## Appendix C

### Employment History of CS Alumni

| Graduation Year | Position | Company |
|---|---|---|
| 1974 | Current: Owner | |
| | | |
| 1975 | Current: Managing Director | |
| | 1. Dean, School Of Business | 1. Tristate University |
| 1978 | Current: Senior Engineer | |
| | 1. Senior Engineer<br>2. Senior Design Engineer | 1. Handera Inc<br>2. Rockwell Collins Avionics |
| 1978 | Current: System Architect | |
| | | |
| 1979 | Current: Software Engineer | |
| | 1. Software Engineer | 1. Mitchell International (2000 - Present) |
| 1980 | Current: Corporate Parts Analyst | |
| | 1. Electrical Cad Support<br>2. Statistical Analyst<br>3. Reliability Research & Warranty Prediction | 1. Same<br>2. Same<br>3. Same |
| 1980 | Current: Senior Staff Systems Engineer | |
| | 1. Systems Engineer<br>2. Maintainability Engineer | 1. Itt<br>2. Itt |
| 1981 | Current: Program Manager | |
| | 1. Program Manager/Project Director<br>2. Partner/COO<br>3. Cto<br>4. Region Director<br>5. Projects Director<br>6. Member Of Technical Staff | 1. Anexsys (Joint Venture, Dissolved Into US Bank)<br>2. Axiom Security<br>3. Fastweb (Acquired By Monster)<br>4. Dmr Consulting Group/Fujitsu<br>5. Mci Systemhouse/Worldcom<br>6. Bell Labs/At&T/Lucent |
| 1981 | Current: Senior Software Developer | |
| | 1. Senior Software Developer<br>2. Principle Consultant | 1. Texas Instruments<br>2. Vivare Inc, |

| | | |
|---|---|---|
| 1981 | Current: Senior Software Developer | |
| | 1. Senior Software Developer<br>2. Principle Consultant<br>3. Principle Consultant<br>4. Technical Architect<br>5. Member, Group Technical Staff | 1. Texas Instruments<br>2. Vivare Inc,<br>3. One, Inc.<br>4. Weblink Wireless<br>5. Texas Instruments |
| 1982 | Current: Embedded Software Engineer 5 | |
| | | |
| 1982 | Current: Principal Project Engineer | |
| | | |
| 1982 | Current: Sr. Proposal/Project Leader | |
| | 1. Project Leader<br>2. Project Manager<br>3. Proposal Leader<br>4. Lead Engineer<br>5. Area Leader<br>6. Developer<br>7. Contract Computer Services<br>8. Process Control Developer | 1. Rockwell<br>2. Rockwell<br>3. Rockwell<br>4. Rockwell<br>5. Rockwell<br>6. GE<br>7. GE<br>8. US Steel |
| 1983 | Current: Senior Consultant | |
| | 1. Senior Consultant<br>2. Senior Analyst<br>3. Director Of Consulting<br>4. Lead Analyst<br>5. Senior Programmer Analyst | 1. Digital Consulting And Software Services<br>2. Mann Diversified Services<br>3. Cgi Consulting<br>4. Camco Products And Services<br>5. Brown-Fintube Company |
| 1983 | Current: Senior Technical Program Manager | |
| | | |
| 1983 | Current: Test Analyst 2 | |
| | 1. Contractor/Test Analyst 2<br>2. Senior Technical Support Engineer<br>3. Principal Validation Engineer<br>4. Contractor/Test Analyst<br>5. Contractor/Tester<br>6. Contractor/Developer<br>7. Developer | 1. Volt Information Systems<br>2. Nsi Software<br>3. Tei, Inc.<br>4. Technisource<br>5. Cap Gemini America<br>6. Cap Gemini America<br>7. Applied Computing Devices |
| 1984 | Current: Data Architect | |
| | 1. Data Architect<br>2. Software Developer<br>3. International Software Support | 1. Standard Register<br>2. Ncr<br>3. Ncr |

| | | |
|---|---|---|
| 1984 | Current: General Partner | |
| | 1. Rural Carriers Assistant | 1. United States Postal Service |
| 1986 | Current: Telemarketing | |
| | 1. Telemarketing<br>2. Grader<br>3. Programmer/Consultant<br>4. Delivery Person<br>5. Roofing Contractor<br>6. Stocker<br>7. Device Driver Programmer<br>8. Database Programmer<br>9. Stocker<br>10. Contract Programmer<br>11. Kernel Developer<br>12. Contract Programmer | 1. 1800Flowers.Com<br>2. Kelly Services<br>3. Thomson Electronics<br>4. Sbc<br>5. Crampton Inc<br>6. Stride Rite<br>7. Infraworks<br>8. The 401K Company<br>9. Lazarus<br>10. Centillion Data Systems<br>11. Network Specialists<br>12. Ameritech (Now Sbc) |
| 1986 | Current: Systems Analyst | |
| | 1. Systems Analyst<br>2. Programmer | 1. Eli Lilly & Company<br>2. Ncr |
| 1987 | Current: Lead 3D Artist | |
| | 1. Application Lab Manager<br>2. Unix Systems Support Director<br>3. Unix Admin<br>4. Vp Unix Support Group<br>5. Member Of Technical Staff<br>6. Engineer | 1. St Microelectronics<br>2. Information Resources Inc<br>3. Vlsi<br>4. Salomon Brothers<br>5. Asic Designs<br>6. Tellabs |
| 1987 | Current: Software Architect | |
| | 1. Technical Project Leader<br>2. Technical Advisor<br>3. Technical Specialist<br>4. Sr Engineer<br>5. Engineer | 1. Cummins, Inc<br>2. Cummins, Inc<br>3. Cummins, Inc<br>4. Cummins, Inc<br>5. Cummins, Inc |
| 1988 | Current: Supervisor - Technology Service Center | |
| | 1. Computer Information Specialist<br>2. Computer Technician<br>3. Computer Technician<br>4. Computer Hardware Specialist | 1. Kansas State University<br>2. Office Products Inc.<br>3. Computerland<br>4. Hutchinson Community College |
| 1989 | Current: Associate Information Consultant | |
| | 1. Systems Architect<br>2. Team Leader<br>3. Database Administrator<br>4. Systems Developer<br>5. Systems Administrator | |

| | | |
|---|---|---|
| 1990 | Current: Application Developer | |
| | | |
| 1990 | Current: Associate Vice President Of Software Development | |
| | 1. Enterprise Architect<br>2. Principal Consultant<br>3. Consultant<br>4. Consultant<br>5. Consultant<br>6. Software Engineer | 1. Bellsouth<br>2. Cingular<br>3. Texas Instruments<br>4. Gte/Verizon<br>5. IBM<br>6. Loral Data Systems |
| 1990 | Current: Consulting It Specialist | |
| | 1. Consulting It Specialist<br>2. J2Ee Architect<br>3. Senior Member Of Technical Staff<br>4. Software Engineer | 1. IBM<br>2. Gemstone Systems<br>3. Knowledge Systems Corp.<br>4. Ascent Logic Corp. |
| 1991 | Current: Director, Professional Services | |
| | 1. Sr. Director, Enterprise Architecture<br>2. Director, Professional Services<br>3. Sr. Director, Technology<br>4. Director, Professional Services<br>5. Project Manager/Technical Architect<br>6. Product Consultant<br>7. Sr. Product Specialist/Product Specialist | 1. Software Company Purchased By Bank Of America<br>2. Software Company Purchased By Bank Of America<br>3. Ipo'D Internet Consulting Company<br>4. Privately Held, Internet Consulting Company<br>5. Privately Held, Internet Consulting Company<br>6. Andersen Consulting (Accenture)<br>7. Andersen Consulting (Accenture) |
| 1991 | Current: Global Compensation And Equity Consultant | |
| | 1. Global Compensation And Equity Consultant<br>2. Department Head, Flexible Staffing<br>3. Intercontinental Regulatory Associate<br>4. Project Lead, Discovery Systems<br>5. Systems Analyst, Discovery Systems<br>6. System Manager, Process Automation | 1. Eli Lilly And Company<br>2. Eli Lilly And Company<br>3. Eli Lilly And Company<br>4. Eli Lilly And Company<br>5. Eli Lilly And Company<br>6. Pfizer Specialty Chemicals |
| 1991 | Current: Software Development Lead | |
| | 1. Software Developer<br>2. Independent Consultant<br>3. Consultant<br>4. Consultant | 1. Jaars/Wycliffe Bible Translators<br>2. Self-Employed<br>3. Esca<br>4. Anderson Consulting |
| 1991 | Current: Staff Software Development Engineer | |
| | | |

| | | |
|---|---|---|
| 1991 | Current: Tech Lead | |
| | 1. Software Engineer / Tech Lead<br>2. Software Engineer<br>3. Software Engineer | 1. Northrop Grumman<br>2. Enviroplan<br>3. ARC |
| 1991 | Current: Vice President, Technical Services | |
| | 1. Resources Director<br>2. Principal Consultant<br>3. Asst Vice President | 1. Pepperweed Consulting<br>2. Pepperweed Consulting<br>3. Conseco |
| 1992 | Current: Director Of Program Management | |
| | | |
| 1992 | Current: Portfolio Manager | |
| | 1. Invertica<br>2. San Vision Technology | 1. CEO<br>2. Cto |
| 1992 | Current: Sr. Network Engineer | |
| | 1. Sr. Network Engineer<br>2. Network Engineer Lead<br>3. Network Engineer<br>4. Captain | 1. Micron Tech, Inc / IS - Service Management<br>2. Micron Tech, Inc / IS - Network Management<br>3. Micron Tech, Inc / IS - Network Services<br>4. U.S. Army |
| 1993 | Current: Senior Manager, Information Technology | |
| | 1. Senior Manager, Network Engineering<br>2. Senior Network Architect<br>3. Network Operations Branch Manager<br>4. Network Engineer | 1. Symantec<br>2. Ntt<br>3. US Air Force<br>4. US Air Force |
| 1993 | Current: Software Designer | |
| | 1. Software Designer | 1. Infinite Software Solutions Inc. |
| 1993 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Senior Software Engineer | 1. Interactive Intelligence, Inc.<br>2. Engineering Animation, Inc. |
| 1993 | Current: Software Engineering Consultant | |
| | | |
| 1993 | Current: System Administrator | |
| | 1. Software Engineer | 1. Command System, Inc. (Now General Dynamic C4S) |

| | | |
|---|---|---|
| 1994 | Current: Database Architect | |
| | 1. Database Architect<br>2. Peoplesoft Developer/Team Lead<br>3. Database Administrator/Peoplesoft System Administrator<br>4. Software Engineer<br>5. Software Applications Engineer | 1. Education Lending Group<br>2. 3M Precision Optics<br>3. 3M Precision Optics/Corning Precision Lens<br>4. Corning Precision Lens/US Precision Lens<br>5. International Centers For Telecommunication Technology |
| 1994 | Current: Software Engineer | |
| | 1. Software Engineer | 1. Motorola |
| 1994 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Teaching Assistant<br>3. Software Engineer | 1. Cisco Systems, Inc.<br>2. University Of Illinois At Urbana-Champaign<br>3. Motorola, Inc. |
| 1995 | Current: Manager | |
| | 1. Manager<br>2. Supervisor<br>3. Software Developer | 1. Beckman Coulter<br>2. Beckman Coulter<br>3. Beckment Coulter |
| 1995 | Current: Staff Engineer | |
| | 1. Staff Engineer | 1. Qualcomm, Inc. |
| 1996 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Software Engineer | 1. Appriss, Inc.<br>2. Rockwell Collins |
| 1997 | Current: Self-Employed | |
| | 1. Director Of Research<br>2. Director Of Research<br>3. Director Of Research | 1. Tumbleweed Communications<br>2. Corvigo, Inc.<br>3. Radiate, Inc. |
| 1997 | Current: Senior Developer | |
| | 1. Senior Developer<br>2. Workstation Manager<br>3. Software Specialist | 1. Idaho Information Consortium<br>2. Rose-Hulman Institute Of Technology<br>3. Rose-Hulman Institute Of Technology |
| 1997 | Current: Senior Manager | |
| | 1. Senior Manager | 1. Manhattan Associates |
| 1997 | Current: Senior Software Engineer | |
| | 1. Senior Software Engineer<br>2. Programmer Analyst | 1. Medplus, Inc.<br>2. Tyson Foods |

| | | |
|---|---|---|
| 1997 | Current: Software Consultant | |
| | 1. Senior Software Engineer<br>2. Software Engineer | 1. Lucent Technologies<br>2. Northrop Grumman Corporation |
| 1998 | Current: Not Reported | |
| | | |
| 1998 | Current: Senior Technical Specialist | |
| | | |
| 1998 | Current: Software Engineer | |
| | 1. Software Engineer | 1. Software Engineering Professionals |
| 1999 | Current: Principal Support Engineer | |
| | 1. Software Engineer | 1. Interactive Intelligence |
| 1999 | Current: Senior Technical Consultant | |
| | 1. Systems Analyst | 1. Made2Manage |
| 2000 | Current: Not Reported | |
| | | |
| 2000 | Current: Application Developer | |
| | | |
| 2000 | Current: Chief Software Engineer | |
| | | |
| 2000 | Current: Chief Technical Officer | |
| | 1. Cto<br>2. Technical Director<br>3. Senior Software Architect<br>4. Software Architect | 1. Total Child Health, Inc. (Baltimore, Md)<br>2. Center For Promotion Of Child Development Through Primary Care (Baltimore, Md)<br>3. The Adrenaline Group, Inc. (Washington, Dc)<br>4. The Adrenaline Group, Inc. (Washington, Dc) |
| 2000 | Current: Senior Engineer | |
| | 1. Senior Engineer/Engineer | 1. Sep |
| 2000 | Current: Software Architect | |
| | 1. Solutions Developer<br>2. Software Developer | 1. Avanade<br>2. Geneer |
| 2000 | Current: Software Design Engineer In Test Lead | |
| | 1. Software Design Engineer In Test | 1. Microsoft |
| 2000 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Software Engineer Associate | 1. Lockheed Martin<br>2. Lockheed Martin |

| | | |
|---|---|---|
| 2000 | Current: Software Integration Engineer | |
| | 1. Chief, Network Implementation Branch<br>2. Network Project Manager<br>3. Crew Commander, Network Operations Center<br>4. Communications Officer<br>5. Officer In Charge, Communications Control Center | 1. Air Force Pentagon Communication Agency<br>2. Air Force Pentagon Communication Agency<br>3. 1St Communications Squadron<br>4. 74Th Air Control Squadron<br>5. 1St Communications Squadron |
| 2001 | Current: Software Engineer | |
| | 1. Software Engineer | 1. Software Engineering Professionals |
| 2001 | Current: Engineer III | |
| | 1. Software Engineer | 1. Honeywell |
| 2001 | Current: Senior Software Architect | |
| | 1. Senior Software Architect<br>2. Senior Consultant | 1. Sterling Creek Software<br>2. Crowe, Chizek |
| 2001 | Current: Senior Software Developer | |
| | 1. Senior Software Developer<br>2. Software Engineer | 1. Columbia House<br>2. Motorola |
| 2001 | Current: Software Design Engineer | |
| | 1. First Job Out Of School At Current Position | |
| 2001 | Current: Software Developer | |
| | 1. Software Developer | 1. Compuware Corporation |
| 2001 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Software Engineer<br>3. Systems Administrator<br>4. Programmer | 1. Evdb, Inc.<br>2. Tomo Software<br>3. Ucsd School Of Engineering<br>4. Bomis |
| 2001 | Current: Staff Engineer | |
| | 1. Staff Engineer<br>2. Senior Consultant<br>3. Software Engineer II | 1. Macaulay Brown<br>2. Accenture<br>3. Northrop-Grumman |
| 2001 | Current: Web Application Developer | |
| | 1. Web Application Developer | 1. Eagan, Mcallister Associates, Inc. |
| 2002 | Current: Project Manager | |
| | 1. Project Manager<br>2. Application Developer | 1. NIC<br>2. NIC |
| 2002 | Current: Embedded Software Engineer Grade 2 | |
| | | |

| | | |
|---|---|---|
| 2002 | Current: Project Coordinator & Developer | |
| | | |
| 2002 | Current: Software Developer | |
| | | |
| 2002 | Current: Software Engineer | |
| | | |
| 2002 | Current: Software Engineer Or Software Consultant | |
| | 1. Software Engineer | 1. Wlasuk, Delporte, And Davis, Inc (Www.Wddinc.Com) |
| 2003 | Current: Not Reported | |
| | | |
| 2003 | Current: Asset Management Intern | |
| | 1. Asset Management Intern<br>2. Graduate Assistant<br>3. Security Coordinator | 1. Las Vegas Valley Water District<br>2. Unlv, Economics Department<br>3. Shindeldecker Oil Company |
| 2003 | Current: Coach | |
| | 1. Coach<br>2. Coach<br>3. Birthday Party Instructor | 1. Cascade Elite Gymnastics<br>2. Gymnastics Plus<br>3. Mid Atlantic Gymnastics |
| 2003 | Current: Programmer | |
| | 1. Graphic Developer | 1. Stagelogic |
| 2003 | Current: Software Developer | |
| | 1. Software Developer | |
| 2003 | Current: Software Engineer | |
| | 1. Software Engineer<br>2. Software Engineer | 1. Quantitative Risk Management<br>2. Northrop Grumman |
| 2003 | Current: Software Engineer Ldp | |
| | 1. Software Engineer Ldp<br>2. Associate Software Engineer Ldp<br>3. Associate Software Engineer | 1. Lockheed Martin<br>2. Lockheed Martin<br>3. Lockheed Martin |
| 2003 | Current: Systems Analyst | |
| | | |
| 2004 | Current: Systems Analyst | |
| | 1. Systems Analyst | 1. Williams Randall Mc |
| 2004 | Current: Assistant Technical Staff | |
| | | |

| | | |
|---|---|---|
| 2004 | Current: Associate Programmer | |
| | 1. Associate Programmer<br>2. Programmer Analyst II<br>3. Programmer Intern | 1. Sigil Games Online, Inc.<br>2. Institute For Creative Technologies, USc<br>3. Institute For Creative Technologies, USc |
| 2004 | Current: Consultant | |
| | | |
| 2004 | Current: Software Engineer | |
| | 1. Software Engineer | 1. Rockwell Collins Inc |
| 2004 | Current: Software Engineer | |
| | | |
| 2004 | Current: Software Engineer | |
| | | |
| 2004 | Current: Systems Engineer | |
| | 1. Systems/Software Engineer | 1. Northrop Grumman |
| 2005 | Current: Software Engineer (Services) | |
| | 1. Software Engineer (Services) | 1. Par3 Communications |
| 2005 | Current: Software Design Engineer | |
| | | |
| 2005 | Current: Software Design Engineer | |
| | 1. Software Design Engineer | 1. Expedia, Inc. |
| Not Reported | Current: Software Developer | |
| | 1. Software Developer | 1. Indiana University |
| Not Reported | Current: It Director | |
| | | |
| Not Reported | Current: Senior Software Engineer | |
| | | |
| Not Reported | Current: Software Engineer | |
| | 1. Technical Support<br>2. Software Developer<br>3. Software Engineer | 1. Caltech Communications<br>2. Wth Engineering<br>3. Software Engineering Professionals |

## Appendix D

### Services CSSE Department can Offer Alumni

| Graduation Year | Comments |
|---|---|
| 1975 | • job career openings |
| 1983 | • A list of CSSE students looking for Summer internships should be pushed out to alumni.<br>• I have been unemployed twice recently due to the market changes. It would have been nice to have a job support network from Rose. |
| 1984 | • well actually none as it would pertain to my profession. My part time jobs either. |
| 1986 | • Re-training. I've never been given an opportunity with companies I've worked with to learn new technologies for the most part. Either I didn't have the time or dropped the ball myself on keeping up with technologies or shift in paradigms. Also, not all of us knew (including me) how to deal in the big corporate environment of which I had no clue. Good thing I knew how to BS. Some etiquette lessons probably wouldn't hurt for some of us red-necks.<br>• Contacts |
| 1987 | • I would like to have an awareness of what issues are facing current faculty and students. What are active research areas? What are the latest changes in emphasis in the curriculum? |
| 1988 | • job search assistance |
| 1990 | • Knowing what other alumni are at my employer or in my geographical area. (R-HIT can do that for the CSSE Dept.) |
| 1991 | • Periodic cash gifts :-) |
| 1992 | • Feedback to changes in CSSE education and new ideas and theories being discussed. Would be interesting to see how educational changes and priorities relate to the business environment. |
| 1993 | • How about a portfolio of graduating students. Like a job site where they could post their resume. It would be nice to see if any graduating students might meet a business need. |
| 1994 | • Annual or semi-annual get-togethers with the faculty to discuss what is being taught and what is being used in the industry. This would allow us the opportunity to know what to expect from recent graduates and it would give the faculty an idea of what we need in new graduates.<br>• Networking with other alums |
| 1997 | • Better communications with alumni, staff and students perhaps as a message discussion forum.<br>• I would like to see summary results of this survey. I would like to have a sense of where others from my "generation" have gone. I would be particularly interested in the salary ranges for my generation and the latest generation.<br>• Would be neat to see the currently taught curriculum to see how it differs from what we were taught. |
| 1999 | • Create a column for the alumni magazine that talks about a new technique or design pattern that hasn't moved from academic to industry yet. |

| | |
|---|---|
| 2000 | • Mailing list? Perhaps there is one that I just don't know about.<br>• Provide opportunities or ideas to further my own intellectual growth in computer science and software engineering.<br>• Masters in Software Engineering program. Access to information on post-graduate study programs available within Rose-Hulman and elsewhere. More interaction with current students and professors on what skills are most important usefulness in industry now than what we were taught 5 years ago. |
| 2001 | • Perhaps a bulletin board service or a forum for alumni to allow them to keep in touch, post interesting projects they are working on and allow them to meet current students online. |
| 2002 | • (Voluntary) Directory of where other CS(SE) graduates are living and working, contact information / email addresses. A voluntary directory such as this would be very helpful.<br>• Career charging forum; technical forums<br>• Access to current and past course offerings to see how the CSSE field and department have developed over the years. Perhaps provide CSSE alumni access to the notes and/or lecture recordings for certain CSSE courses. Alumni can then audit the courses (may prove beneficial for new or experimental courses). Keeping alumni up to date on open CSSE course presentations. Alumni may find it interesting/beneficial to attend. |
| 2003 | • employment services<br>• An update on current classes.<br>• Professional opportunity contacts; some job communication line just kept in the CSSE department.<br>• Better professor websites. There have been many times when I remembered something I learned (but then forgot) and tried to look it up using the course web pages, but could not find the information or leads.<br>• A thought just popped into my head... What about a discussion board for alumni to discuss CSSE problems they are dealing with? Professors could/should be members that could provide advice alongside other CSSE alumni. Perhaps there's already a newsgroup for this in the rhit.cs somewhere?? |
| 2004 | • A better career networking website.<br>• A listing of companies that other alumni work for and how long they've been there. It's intimidating to try to find a good company to work for on your own. I haven't actually checked if you provide this information currently, but it's something I would find helpful in the future.<br>• A newsletter with the current events of Rose and the department<br>• yearly update of the students, staff, and faculty in the CSSE Department, short description of professor's experience in his/her sabbatical<br>• Contact between alumni message forum newsletter |
| 2005 | • Contact info, department updates (what's going on in the dept.)<br>• They should move to Washington and offer a Masters of Software Engineering. |
| Not Reported | • a newsletter would be good enough<br>• Maybe send out curriculums occasionally so that alumnus can give feedback on how each would or would not be beneficial in the profession he/she has. |

## Appendix E

### Opportunities for Alumni to Give to Current Students

| Graduation Year | Comments |
|---|---|
| 1972 | • Job experience |
| 1975 | • Internships |
| 1976 | • Real-world experience in a variety of environments |
| 1979 | • The opportunities are there, but I live too far away and am way too busy with career and family to be able to offer any time right now. |
| 1980 | • Mentoring programs |
| 1981 | • counsel potential RHIT freshman (i.e., high schoolers), career counseling for seniors, possible resource for project work |
| 1982 | • Relating real-world experience - applying CSSE education to process control systems, expanding the horizon for career choices |
| 1983 | • I cannot recall being contacted for recruiting or internship positions.<br>• I have hired graduates (as Research Assistants) coming to the University of Illinois for graduate school. |
| 1984 | • Job references<br>• nothing really---except to let them know not everyone is an engineer so not everyone can figure out what they are talking about. And not everything that's worth making is financially breath taking so sometimes do things that might not make you rich. |
| 1986 | • I have quite a few contacts being an ex-consultant and working with many companies as they approach graduation.<br>• Shadowing experiences, Career presentations |
| 1987 | • I would like to know more. I would be glad to help. I'm aware of senior design projects. I presume you might have some alumni come in to speak to a class or for an evening seminar. I would be willing to do something like this, but I don't know what you need. |
| 1989 | • Presenting topics in class from real-world experiences |
| 1990 | • IBM Academic Initiative -- http://www.developer.ibm.com/university/scholars/ -- http://www-128.ibm.com/developerworks/blogs/dw_blog_comments.jspa?blog=392&entry=78809<br>• The usual. Job referral. Project Ideas. Professional Guidance. A Dose of Reality. |
| 1991 | • Proposing senior projects. Volunteering to give a guest lecture. Sharing with Christian students how they can use their technical skills for the Kingdom of God.<br>• Sharing "real-world" corporate Software Development approaches & strategies, and how the RHIT education can provide significant leadership in this space. |
| 1992 | • Computer science grads looking for work on Wall Street should contact me. I think we need more people with good problem solving and analytical skills.<br>• I believe the ability to provide feedback and mentorship to students and to help guide them in their education. Feedback to the professors on trends and directions might also be of benefit. |

| | |
|---|---|
| 1993 | • I am in a position to hire students, both as interns and full-time.<br>• To provide real-world projects on an internship basis so that students can get exposed to real-world requirements. |
| 1994 | • I will be going to Graduate school within the next quarter or two. After I have completed my schooling, I would ultimately like to teach at Rose one day.<br>• Providing leadership for senior projects. |
| 1995 | • Give discussions on the software development process and how it is being applied in real life. Stress what parts of the development process we found most useful in commercial practice compared to my perceptions when I was at school. (For example, I've found that code reviews are important, but weren't really brought up in schoolwork).<br>• I think my biggest contribution is as a networking connection. |
| 1997 | • Alumni could give seminars to students in which they relate their professional experiences, describe the work they have done, and answer questions. Seminars could also be overviews of new research or cool technology related to the field. The point would be to give students an idea of what's going on in the industry, give them examples of professionals giving presentations, and to utilize the knowledge of alumni to help educate students in areas that are maybe less academic.<br>• I think the senior projects are very valuable in providing students opportunities to experience the use of their education.<br>• Lining up senior projects.<br>• Without better communications, any speculation on the current students needs is unreliable. |
| 1999 | • While the software engineering courses may have said that most of the cost of the product occurs after the initial development, that isn't reinforced by the "CS project experience". I am currently in my companies support department, and I can see this first hand, and I could talk about it. |
| 2000 | • An opportunity exists for guest speaking in a class about what I have learned in the past 5 years since graduation. Knowing what I do now, I would like to share those things that I practiced that were successful and things that I wished I had done or should not have done in order to be even more successful. Also, there is an opportunity for senior projects with industry to foster relationships and hiring students after graduation.<br>• Hopefully, a senior project idea and opportunity ;)<br>• Mentoring after graduation<br>• Provide more realistic opportunities to mentor and/or teach real-world, practical software engineering to students. As it stands, the only opportunity to provide such help is as a Senior Project sponsor, but the time and resource requirements for that are extremely large. Smaller packets of "help" would be more available.<br>• Senior Projects |

| | |
|---|---|
| 2001 | • Alumni could be mentors for current students, either informally or formally, as long as there's enough alumni and students interested in something like this. Also, giving money to Rose-Hulman by alumni benefits current students.<br>• Employment opportunities, experiences in the workforce, social networking, etc.<br>• Giving students a better understanding of real world problems and situations, which is something the current curriculum is sorely lacking.<br>• Guest lectures. Insight into current state of affairs for CSSE graduates. Approaches to gaining employment.<br>• Senior Projects - I really enjoyed mine, and hope to sponsor one in the future! Summer Internships - We have had several RHIT students as interns<br>• Sponsoring a senior project (something I've been agitating to do at work :))<br>• Tips on how to be a good employee, and a good contributor to a team. |
| 2002 | • Ability to discuss graduate school opportunities and what courses / actions should be taken as an undergraduate to better prepare for a research career in computer science.<br>• Alumni can provide a real-world look in the various areas of computer science. Perhaps being a guest lecturer for a class would bring some insight to the students.<br>• I would love the chance to sponsor a software engineering team or a senior project team.<br>• Our company has a history of hiring RHIT graduates. Our excellence causes them to continue going back to RHIT for almost all new hires. |
| 2003 | • Advice. Internships. Perspective on why their learning is important, especially in software engineering.<br>• give money and material<br>• I could mentor, but with my current state of employment, financial gifts are out of the question.<br>• I would like to be an advisor to undergraduate students. I am not sure that there is a system set up for that.<br>• Mostly just advice in my particular field<br>• Original research or development projects for classes, money.<br>• To show them that there is a world of CS that is way beyond what they learn in the classroom, that not all CS graduate go on to be code monkeys or grad students. |
| 2004 | • I do some recruiting for my company, and that also allows me to talk with classes about practical applications of software engineering<br>• I have direct experience using my degree and recent experience attending classes. My knowledge about which courses are or are not valuable to me could help current students plan their academic path.<br>• I have received offers to come back and talk about my experiences. Obviously the professors want us to share our knowledge to the current students.<br>• Mentorship. Reality-check. |
| 2005 | • Give alums the chance to come back to relate experiences at Rose to the real world. It helps to show the students that all their hard work will pay off. Also give students the chance to ask alumni questions.<br>• List of the things I wish I had known before a graduated, job opportunities, and monetary donations. Talks or demonstrations of cool stuff I know about my industry.<br>• Refer them to jobs at Microsoft. |

| | |
|---|---|
| Not Reported | • - donations - senior projects (subject to approval by employer)<br>• Give curriculum feedback. Alumni could give talks to students on what their jobs entail and how different items from the curriculum are being used. |

# Crosstabs

**Case Processing Summary**

| | Cases | | | | | |
|---|---|---|---|---|---|---|
| | Valid | | Missing | | Total | |
| | N | Percent | N | Percent | N | Percent |
| New CSSE Group * Prepare for Success | 124 | 95.4% | 6 | 4.6% | 130 | 100.0% |
| New CSSE Group * Overall Value | 124 | 95.4% | 6 | 4.6% | 130 | 100.0% |
| New CSSE Group * Recommend to Others | 122 | 93.8% | 8 | 6.2% | 130 | 100.0% |
| New CSSE Group * Overall Satisfaction | 123 | 94.6% | 7 | 5.4% | 130 | 100.0% |

**New CSSE Group * Prepare for Success Crosstabulation**

| | | | Prepare for Success | | | | |
|---|---|---|---|---|---|---|---|
| | | | Poorly | Fair | Well | Very well | Total |
| New CSSE Group | Prior to 2001 | Count | 1 | 6 | 43 | 33 | 83 |
| | | % within New CSSE Group | 1.2% | 7.2% | 51.8% | 39.8% | 100.0% |
| | 2001-2003 | Count | 2 | 3 | 13 | 11 | 29 |
| | | % within New CSSE Group | 6.9% | 10.3% | 44.8% | 37.9% | 100.0% |
| | 2004-2005 | Count | 0 | 4 | 6 | 2 | 12 |
| | | % within New CSSE Group | .0% | 33.3% | 50.0% | 16.7% | 100.0% |
| Total | | Count | 3 | 13 | 62 | 46 | 124 |
| | | % within New CSSE Group | 2.4% | 10.5% | 50.0% | 37.1% | 100.0% |

**New CSSE Group * Overall Value Crosstabulation**

| | | | Overall Value | | | | |
|---|---|---|---|---|---|---|---|
| | | | Poorly | Fair | Well | Very well | Total |
| New CSSE Group | Prior to 2001 | Count | 1 | 12 | 39 | 31 | 83 |
| | | % within New CSSE Group | 1.2% | 14.5% | 47.0% | 37.3% | 100.0% |
| | 2001-2003 | Count | 2 | 3 | 13 | 11 | 29 |
| | | % within New CSSE Group | 6.9% | 10.3% | 44.8% | 37.9% | 100.0% |
| | 2004-2005 | Count | 2 | 2 | 6 | 2 | 12 |
| | | % within New CSSE Group | 16.7% | 16.7% | 50.0% | 16.7% | 100.0% |
| Total | | Count | 5 | 17 | 58 | 44 | 124 |
| | | % within New CSSE Group | 4.0% | 13.7% | 46.8% | 35.5% | 100.0% |

**New CSSE Group * Recommend to Others Crosstabulation**

| | | | Recommend to Others | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Not at all likely | Not very likely | Neither likely nor unlikely | Somewhat likely | Very likely | Total |
| New CSSE Group | Prior to 2001 | Count | 2 | 2 | 1 | 25 | 51 | 81 |
| | | % within New CSSE Group | 2.5% | 2.5% | 1.2% | 30.9% | 63.0% | 100.0% |
| | 2001-2003 | Count | 0 | 0 | 5 | 9 | 15 | 29 |
| | | % within New CSSE Group | .0% | .0% | 17.2% | 31.0% | 51.7% | 100.0% |
| | 2004-2005 | Count | 0 | 1 | 2 | 6 | 3 | 12 |
| | | % within New CSSE Group | .0% | 8.3% | 16.7% | 50.0% | 25.0% | 100.0% |
| Total | | Count | 2 | 3 | 8 | 40 | 69 | 122 |
| | | % within New CSSE Group | 1.6% | 2.5% | 6.6% | 32.8% | 56.6% | 100.0% |

**New CSSE Group * Overall Satisfaction Crosstabulation**

| | | | Overall Satisfaction | | | | |
|---|---|---|---|---|---|---|---|
| | | | Somewhat dissatisfied | Neither satisfied nor dissatisfied | Somewhat satisfied | Very satisfied | Total |
| New CSSE Group | Prior to 2001 | Count | 1 | 2 | 37 | 42 | 82 |
| | | % within New CSSE Group | 1.2% | 2.4% | 45.1% | 51.2% | 100.0% |
| | 2001-2003 | Count | 1 | 2 | 14 | 13 | 30 |
| | | % within New CSSE Group | 3.3% | 6.7% | 46.7% | 43.3% | 100.0% |
| | 2004-2005 | Count | 1 | 1 | 7 | 2 | 11 |
| | | % within New CSSE Group | 9.1% | 9.1% | 63.6% | 18.2% | 100.0% |
| Total | | Count | 3 | 5 | 58 | 57 | 123 |
| | | % within New CSSE Group | 2.4% | 4.1% | 47.2% | 46.3% | 100.0% |

# Frequencies

**Statistics**

New CSSE Group

| | | |
|---|---|---|
| N | Valid | 126 |
| | Missing | 4 |

**New CSSE Group**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | Prior to 2001 | 84 | 64.6 | 66.7 | 66.7 |
| | 2001-2003 | 30 | 23.1 | 23.8 | 90.5 |
| | 2004-2005 | 12 | 9.2 | 9.5 | 100.0 |
| | Total | 126 | 96.9 | 100.0 | |
| Missing | System | 4 | 3.1 | | |
| Total | | 130 | 100.0 | | |

# NPar Tests

# Kruskal-Wallis Test

**Ranks**

| | New CSSE Group | N | Mean Rank |
|---|---|---|---|
| Importance: Ethics | Prior to 2001 | 83 | 64.06 |
| | 2001-2003 | 30 | 64.38 |
| | 2004-2005 | 12 | 52.21 |
| | Total | 125 | |
| Contribution: Ethics | Prior to 2001 | 83 | 59.07 |
| | 2001-2003 | 29 | 70.64 |
| | 2004-2005 | 12 | 66.54 |
| | Total | 124 | |

**Ranks**

| | New CSSE Group | N | Mean Rank |
|---|---|---|---|
| Importance: Contemporary Issues | Prior to 2001 | 83 | 59.55 |
| | 2001-2003 | 30 | 73.97 |
| | 2004-2005 | 12 | 59.42 |
| | Total | 125 | |
| Contribution: Contemporary Issues | Prior to 2001 | 83 | 58.54 |
| | 2001-2003 | 29 | 70.48 |
| | 2004-2005 | 12 | 70.63 |
| | Total | 124 | |
| Importance: Global | Prior to 2001 | 82 | 61.79 |
| | 2001-2003 | 30 | 65.15 |
| | 2004-2005 | 12 | 60.75 |
| | Total | 124 | |
| Contribution: Global | Prior to 2001 | 82 | 56.65 |
| | 2001-2003 | 29 | 68.97 |
| | 2004-2005 | 12 | 81.71 |
| | Total | 123 | |
| Importance: Culture | Prior to 2001 | 82 | 64.05 |
| | 2001-2003 | 30 | 56.77 |
| | 2004-2005 | 12 | 66.25 |
| | Total | 124 | |
| Contribution: Culture | Prior to 2001 | 82 | 61.84 |
| | 2001-2003 | 29 | 61.69 |
| | 2004-2005 | 12 | 63.83 |
| | Total | 123 | |
| Importance: Teams | Prior to 2001 | 82 | 65.01 |
| | 2001-2003 | 30 | 52.50 |
| | 2004-2005 | 12 | 70.38 |
| | Total | 124 | |
| Contribution: Teams | Prior to 2001 | 82 | 54.74 |
| | 2001-2003 | 29 | 73.26 |
| | 2004-2005 | 12 | 84.42 |
| | Total | 123 | |
| Importance: Communication | Prior to 2001 | 81 | 61.17 |
| | 2001-2003 | 30 | 63.98 |
| | 2004-2005 | 12 | 62.67 |
| | Total | 123 | |

**Ranks**

| | New CSSE Group | N | Mean Rank |
|---|---|---|---|
| Contribution: Communication | Prior to 2001 | 81 | 58.00 |
| | 2001-2003 | 29 | 67.59 |
| | 2004-2005 | 12 | 70.42 |
| | Total | 122 | |
| Importance: Problem-Solving | Prior to 2001 | 81 | 62.44 |
| | 2001-2003 | 30 | 61.40 |
| | 2004-2005 | 12 | 60.50 |
| | Total | 123 | |
| Contribution: Problem-Solving | Prior to 2001 | 81 | 62.52 |
| | 2001-2003 | 29 | 62.43 |
| | 2004-2005 | 12 | 52.33 |
| | Total | 122 | |
| Importance: Interpreting Data | Prior to 2001 | 82 | 62.26 |
| | 2001-2003 | 30 | 63.70 |
| | 2004-2005 | 12 | 61.17 |
| | Total | 124 | |
| Contribution: Interpreting Data | Prior to 2001 | 82 | 63.67 |
| | 2001-2003 | 29 | 65.72 |
| | 2004-2005 | 12 | 41.58 |
| | Total | 123 | |
| Importance: Experiments | Prior to 2001 | 82 | 63.91 |
| | 2001-2003 | 30 | 59.92 |
| | 2004-2005 | 12 | 59.29 |
| | Total | 124 | |
| Contribution: Experiments | Prior to 2001 | 82 | 65.14 |
| | 2001-2003 | 29 | 61.16 |
| | 2004-2005 | 12 | 42.58 |
| | Total | 123 | |
| Importance: Design | Prior to 2001 | 80 | 62.98 |
| | 2001-2003 | 29 | 58.03 |
| | 2004-2005 | 12 | 55.00 |
| | Total | 121 | |
| Contribution: Design | Prior to 2001 | 66 | 47.05 |
| | 2001-2003 | 20 | 46.93 |
| | 2004-2005 | 8 | 52.63 |
| | Total | 94 | |

**Test Statistics[a,b]**

| | Importance: Ethics | Contribution: Ethics | Importance: Contemporary Issues | Contribution: Contemporary Issues | Importance: Global | Contribution: Global | Importance: Culture | Contribution: Culture |
|---|---|---|---|---|---|---|---|---|
| Chi-Square | 1.389 | 2.609 | 3.882 | 3.327 | .235 | 7.216 | 1.108 | .040 |
| df | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Asymp. Sig. | .499 | .271 | .144 | .189 | .889 | .027 | .575 | .980 |

**Test Statistics[a,b]**

| | Importance: Teams | Contribution: Teams | Importance: Communication | Contribution: Communication | Importance: Problem-Solving | Contribution: Problem-Solving | Importance: Interpreting Data | Contribution: Interpreting Data |
|---|---|---|---|---|---|---|---|---|
| Chi-Square | 4.782 | 12.538 | .192 | 2.902 | .059 | 1.406 | .062 | 5.476 |
| df | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Asymp. Sig. | .092 | .002 | .909 | .234 | .971 | .495 | .970 | .065 |

**Test Statistics[a,b]**

| | Importance: Experiments | Contribution: Experiments | Importance: Design | Contribution: Design |
|---|---|---|---|---|
| Chi-Square | .400 | 5.049 | 1.011 | .372 |
| df | 2 | 2 | 2 | 2 |
| Asymp. Sig. | .819 | .080 | .603 | .830 |

a. Kruskal Wallis Test

b. Grouping Variable: New CSSE Group

## Tables

| | New CSSE Group | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Prior to 2001 | | | 2001-2003 | | | 2004-2005 | | |
| | Mean | Median | Std Deviation | Mean | Median | Std Deviation | Mean | Median | Std Deviation |
| Importance: Ethics | 4.13 | 4.00 | 1.00 | 4.17 | 4.00 | .91 | 3.83 | 4.00 | 1.03 |
| Contribution: Ethics | 3.39 | 4.00 | 1.10 | 3.79 | 4.00 | .94 | 3.67 | 4.00 | .89 |
| Importance: Contemporary Issues | 2.88 | 3.00 | 1.11 | 3.33 | 3.00 | 1.30 | 2.83 | 3.00 | .94 |
| Contribution: Contemporary Issues | 3.14 | 3.00 | 1.13 | 3.55 | 4.00 | .99 | 3.58 | 3.50 | .90 |
| Importance: Global | 2.84 | 3.00 | 1.25 | 2.97 | 3.00 | 1.19 | 2.83 | 2.50 | 1.40 |
| Contribution: Global | 2.39 | 2.00 | .97 | 2.76 | 3.00 | .95 | 3.17 | 3.00 | 1.11 |
| Importance: Culture | 2.98 | 3.00 | 1.17 | 2.70 | 3.00 | 1.18 | 3.08 | 3.00 | 1.51 |
| Contribution: Culture | 2.62 | 3.00 | .92 | 2.72 | 2.00 | 1.10 | 2.83 | 2.50 | 1.11 |
| Importance: Teams | 4.61 | 5.00 | .68 | 4.17 | 5.00 | 1.12 | 4.58 | 5.00 | 1.00 |
| Contribution: Teams | 3.94 | 4.00 | .88 | 4.41 | 5.00 | .73 | 4.67 | 5.00 | .65 |
| Importance: Communication | 4.51 | 5.00 | .73 | 4.60 | 5.00 | .56 | 4.42 | 5.00 | 1.00 |
| Contribution: Communication | 3.74 | 4.00 | .85 | 4.00 | 4.00 | .80 | 4.08 | 4.00 | .67 |
| Importance: Problem-Solving | 4.53 | 5.00 | .74 | 4.47 | 5.00 | .90 | 4.42 | 5.00 | .90 |
| Contribution: Problem-Solving | 4.68 | 5.00 | .54 | 4.66 | 5.00 | .61 | 4.33 | 5.00 | .98 |
| Importance: Interpreting Data | 4.16 | 4.00 | .90 | 4.20 | 4.00 | .89 | 4.17 | 4.00 | .83 |
| Contribution: Interpreting Data | 4.41 | 5.00 | .77 | 4.48 | 5.00 | .69 | 3.92 | 4.00 | .79 |
| Importance: Experiments | 3.13 | 3.00 | 1.21 | 3.03 | 3.00 | 1.19 | 3.00 | 3.00 | 1.41 |
| Contribution: Experiments | 3.85 | 4.00 | .88 | 3.76 | 4.00 | .83 | 3.42 | 3.00 | .51 |
| Importance: Design | 4.34 | 5.00 | 1.03 | 4.21 | 5.00 | 1.05 | 3.92 | 4.50 | 1.51 |
| Contribution: Design | 3.32 | 3.00 | .77 | 3.35 | 3.00 | .67 | 3.38 | 4.00 | 1.06 |

**DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING**
**Rose-Hulman Institute of Tetchnology**
**Board of Advisors**
**11/30/2005**

| | NAME | SPOUSE | ALUM | ON SINCE | EMAIL | COMPANY | TITLE | ADDRESS | PHONE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Larry Alldredge | none | 1982 | 2002 | lalldredge@yahoo.com | Quallcomm, Inc. | Vice President | 5775 Morehouse Drive, San Diego, CA 92121-1714 | (858) 658-2993 |
| 2 | Michael Bayne | none | 1994 | 2000 | mdb@samskivert.com | Go2Net, Inc. | Chief Architect | 601 4th St., Apt 217, San Francisco, CA 94107-1635 | (415) 615-0515 (H) |
| 3 | Robert Cannon | Jaye | No | 2001 | cannon@sc.edu | | Professor Emeritus of Computer Science, Dept. of Computer Science & Engineering | 48 Old Laurel Dr., Laurel Park, NC 28739-4998 | (803) 777-6853 (W)<br>(801) 751-3586 (FAX) |
| 4 | Brooke Chenoweth | none | 2002 | 2001 | bchenowe@cs.indiana.edu | | | 3506 Mackland Ave., NE., Albuquerque, NM 87106-1219 | ((H) |
| 5 | Conn, Richard L. | none | 1976 | 2003 | rickconn7@msn.com | | Certified Software Development Professional | PMB 208, 2090 Baker Road, Suite 304, Kennesaw, GA 30144 | (678) 521-3440 (FAX) |
| 6 | Todd Eigenschink | Darla | 1993 | 2000 | todd@tekinteractive.com | Tek Interactive Group, Inc. | System Administrator, Senior Partner | P.O. Box 8368, Ft. Wayne, IN 46898-8368 | (260) 459-2521 x204 |
| 7 | Bryan C. Everly | Judy | No | 2000 | bryan@peoplestrategy.com | People Strategy LLC | | 8942 Sawmill Ct., Indianapolis, IN 46236-9171 | (317) 823-0339 (FAX) (317) 823- |
| 8 | Dennis Frailey | | No | 2000 | djfrailey@raytheon.com | Raytheon Company | Principal Fellow | 5000 Avenue K Apt. 2516 Plano, TX 75074-3045 | (972) 344-7701 (FAX) |
| 9 | Samuel D. Franklin | none | 1998 | 2000 | skip@acm.org<br>sfranklin@crowechizek.com | Crowe Chizek & Co., LLP | Senior Consultant | Crowe Chizek, 300 Riverfront Plaza Bldg., 55 Campau Ave., Grand Rapids, MI 49503-2650 | (616) 774-0774 x 446 |
| 10 | Gilbert, Jeffrey S. | Deanna | 1985 | 2005 | jsgilbert@sep.com | SEP, Inc. | President | 11611 N. Meridian St., 8th Floor, Carmel, IN 46032 | W (317) 843-1640<br>(317) 843-1642 (FAX) |
| 11 | Joel F. Klein | Leyla | 1994 | 2000 | jfklein@wolfram.com | Wolfram Research, Inc. | Software Engineer | 713 So. Walnut St., Urbana, IL 61801-4245 | (217) 337-0248 |
| 12 | Carl Murray | Judy | No | 2000 | ckmurray@beckman.com | Beckman Coulter, Inc., Sagian Operations | Manager, Research and Development Sagian Operations | Beckman Coulter, Inc., 7451 Winton Drive, Indianapolis, IN 46268-5103 | (317) 808-4213 (W) |
| 13 | Jane Prey | | No | 2005 | jprey@microsoft.com | Microsoft Research | Academic Innovations Manager, | Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 | O (425) 703-5485<br>cell (540) 908-9056 |
| 14 | Martin Schray | | No | 2003 | mschray@microsoft.com | Microsoft Corporation | Academic Developer Evangelist, US-Developer-ISG-Education | 3025 Highland Parkway, Suite 300, Downers Grove, IL 60515-5506 | (630) 876-5113 (W)<br>(801) 697-5113 (FAX) |
| 15 | Mark Shaw | Catherine | No | 2002 | mark.shaw@motorola.com | Motorola | System Engineer, Network Management | 1881 W. Lantana Ave. Chandler, AZ 85248-2173 | (480) 821-4901 [H] |
| 16 | John Simon E. | Linda | 1989 | 2000 | john.simon@lilly.com | Eli Lilly and Company | Lead Architect for Human Resources, Architecture and Data Strategy | Lilly Corporate Center, Indianapolis, IN 46285 | (317) 276-1676 (W)<br>(317) 433-8790 (FAX) |
| 17 | Subbert, Nathan P. | | Yes | 2004 | npsubber@rockwellcollins.com | Rockwell Colins | Senior Software Engineering, ASIC Engineering, Commercial Systems Engineering | 400 Collins Road NE, Cedar Rapids, IA 52498-0001 | (319) 295-1638 (W)<br>(319) 295-3188 (FAX) |

**CSSE ADVISORY BOARD MEETING**
**APRIL 29, 2005**

Board members present: Bob Cannon, Rick Conn, Todd Eigenschink, Dennis Frailey, Jeff Gilbert, John Simon, and Nathan Subbert.

Faculty present: Claude Anderson, Mark Ardis, Don Bagert, Steve Chenoweth, Archana Chidanandan, Cary Laxer, J. P. Mellor, Larry Merkle, David Mutchler, and Mike Wollowski

Staff present: Lynn Degler

Board members absent: Larry Alldredge, Michael Bayne, Brooke Chenoweth, Bryan Everly, Skip Franklin, Joel Klein, Carl Murray, Jane Prey, Martin Schray and Martin Shaw

Absent: Salman Azhar

Handouts:

- Agenda
- CSSE Annual Report to the Board of Trustees
- 2015 Forum
- ABET
- Final Report of the Committee to Study the MSE Degree
- Summary of feedback on the CSSE 2015 discussion
- CSSE Curricula
- Board of Advisors Recommendations
- Software Engineering Courses
- Board members list

## Call to Order and Introductions

Presented by Cary Laxer

Cary thanked the Board for taking the time out of their busy work schedules to support the department. Introductions were made by all.

## State of the Department Report

By Cary Laxer

Cary distributed the CSSE Annual Report to the Board of Trustees.

Cary reviewed the Department of Computer Science and Software Engineering Annual Report to the Rose-Hulman Board of Trustees. The year for the Department has been very active. We have begun preparations in order to put our two programs, Computer Science and Software Engineering, up for ABET accreditation. Larry Merkle is the Lead Author on the Computer Science self-study and Don Bagert is the Lead Author on the Software Engineering self-study.

The first draft of those have already been submitted to the Dean even though the ABET visit will not be until fall 2006. Frank Young will take a first look at those in May 2005 and provide feedback in order to give us time in the summer to make corrections. In the fall Bob Cannon and Dennis Frailey will be returning to campus to be external evaluators before we submit the self-studies to ABET. Those reports are not ready for anyone to read yet. We have developed objectives, assessment plans and assessment reports for every course we teach in the department. Once these first steps are done there will only be minor modifications that need to done for each. Our students continue to excel. A student group was able to work with students from Uppsala University in Sweden. We have been involved in a programming contest at Carnegie Mellon taking first place. Four female students participated in a Games 4 Girls competition taking second place and winning a $500 scholarship each with the department getting $1,000. Six outstanding theses are being researched. Their posters are on display in the department. We still attract quality projects for senior projects. Our faculty is heavily involved with professional development. We have a new operating systems and computer security lab. Last fall the department hosted the WWW@10 conference on campus attracting several well known speakers from Europe and the United States who gave invited talks. We also had referred papers as well. Our students were enticed to attend the conference and a large majority attended. A course that Cary is team teaching with Bill Pickett, Professor of History, entitled The History of Computing grew out of that. John asked if this course could be used as a Humanities elective for CS. Cary responded that it could. The CSSE is a great department.

We have interviewed five people and made three offers. All three excepted that were made offers to. This has been very time consuming. We have three new faculty joining the department next year due to sabbaticals starting. They are Matthew Boutell, University of Rochester; Curtis Clifton, Iowa State University and Lisa Kaczmarczyk, the University of Texas at Austin. Salman is leaving the department but hopes to return at a later time.

The seniors are still doing well on placement with 87% of the CS majors placed. The highest offer given so far to a senior has been to a CS major from Microsoft. The average offer to a CSSE major is $50,000. One of the Barry M. Goldwater Scholarships has been awarded to a CS major, Angela Smiley. Scott Loughmiller and Jeff Ready, 1996 alumnus, were recently named outstanding young alumni. David is doing a course on robotics and teamwork. The new Software Engineering degree is popular.

**Jack Midgley, President of Rose-Hulman**

President Midgley thanked everyone for coming and appreciates everything that is done for the department.

The President summarized his report on campus evolution. The budget for next year reflects a real academic priority. The tuition increase is being held down this year to 3.9%; the lowest increase in ten years. Six new faculty have been hired for next fall. The Dean's equipment budget has been increased to $600,000. The financial aid budget for next year will be the highest ever. Some things have been done around the health care plan to shift more of the cost to the employees. The number one cost drivers are financial aid and health insurance premiums. Six per cent was taken out of the nonacademic non-salary overhead. No positions were eliminated at

Rose nor has anyone lost their job. We are also involving Rose-Hulman Ventures by trying to pull this back more into the academic program. We found that as we worked on the Ventures program the model of taking equity positions and startups as a way of generating project opportunities is a really expensive way to create undergraduate projects. There is no more an expensive way to do it than that. Also, in the current economic environment, particularly given the Venture capital environment it was just not smart for us to continue to rely on that so we are involving that model bringing Ventures closer to the academic program placing it under the supervision of the Dean and involving faculty in a redesign of the program so that it can be the center piece of project based education at Rose-Hulman. Projects have always been important and they are going to keep being important.

The redesigned Ventures has a couple of components that are being developed by faculty. One is having a project management office. This will be the hub for drawing project activity from established and startup companies into Rose-Hulman. The second piece is called collaboratory areas. What we are trying to do there is overcome what ever disadvantages attributed to us by our remote location. Here in Terre Haute we don't have 150 Fortune fifty companies to generate projects for our students but if we can use the collaborative technologies that are now emerging and, we hope the CSSE Advisory Board can help with this, we can be where we can't be physically. We can bring clients in, we can share development work, we can allow our students to experience the kind of distributed development environments and work environments that are characteristic of professional practices in the world today. Cary is trying to figure out with a team of faculty how to do that. One of the models that we talked about is a three pointed project where we have a remote corporate client, a development or software engineering team in Terre Haute (students and faculty). and maybe other students and faculty in a third location linked together in a real-time or near real-time environment. Some of the evolutions that are being done on campus are to get us more focused on academics, to bring project based education closer to the curriculum and to build collaborative capabilities so we can be where we want to be. Even the changes that have been made this year in the athletic program reflect academics first. We changed athletic conferences moving from the Southern Athletic conference to the Heartland athletic conference from Texas to Indiana. That is going to save us about $100,000 in travel cost and also going to save however many days of student time where they are not just wasting time flying back and forth to go play a game. We're going to play the schools that are our neighbors and our traditional rivals. We can't get that done next year because it takes two years to change conferences but we have already made the announcement.

We are entering a period unlike any period in private higher education that has not been seen before. The students that we want, the 1320s and 1350s SAT scores students, are ready to do math, science and engineering but we have a couple of issues here.

1. That population is shrinking both nationally and in Indiana. Half of our class comes from Indiana so when you look at a shrinking pie nationally and you look at the extra shrink that we get in Indiana, we have some issues there.

2. We have a fierce competitor in Purdue University. They are wide awake. They slept for a long time but no longer. They start at a comprehensive tuition rate of $6,000 for in states while we are at $27,500. There average starting salary is $52,000; our average starting

salary is $53,000. You can't make those numbers balance so we have to do other things. Purdue is calling kids who are cross supplied to Rose-Hulman and saying when you get your financial aid offer from Rose call them. They are basically waving tuition. They are going after the student population that we are seeking. Case Western is doing the same thing. The Admissions folks haven't seen as much of that activity from Carnegie Mellon but the President thinks it's happening there also. The pie is shrinking and so the competition is heating up. We do not have the financial aid profile that these other schools have. We have got to get that fixed or we are not going to get the students to teach. That's one form of competitive pressure. A second form is that the jobs that our students want are not primarily in Indiana. We are going to have to get very creative, very fast in making clear that we have an offering here that is unique and that it is unmatched in any other environment. That's why projects like collaboratory are so vital. We have to get to work on financial aid.

3. 2015. We don't have at the moment a comprehensive picture of what we want Rose-Hulman to look like in 2015. A few months back we started a project called Conversation about the Future. We have invited our faculty, staff, students, alumni, trustees, community members, everyone just to share hopes and dreams about what the Institution might be. That kind of turns peoples attitudes toward the future. Beginning over the summer and continuing next year we will build the 2015 plan for the Institution. The physical transformation of the campus, the academic programs, the endowment building and the fund raising are what we need to do. Hopefully, this process will be done by spring 2006 and we can be fund raising in the fall of 2006/2007. We need help from the Advisory Board on this. We need a lot of outside thinking. We need corporate advice, we need to really make sure that we are aggressively looking out ten years and thinking about what we need to do to be the best in the world in undergraduate math and science engineering education. If we can do that then the future will continue to be very very bright.

Dennis asked if in 1985 if anyone did a 2005 thing. Jack replied that in 1991 a project was done that was the start of the Vision to be the Best campaign. There was no real time line on it. This campaign ended last June. At the present time there is no long range plan for the Institution. An architect came to campus and talked to the faculty, staff and students and looked at the buildings. They recommended that the Logan Library, Crapo Hall and BSB be replaced. They liked Deming, although they think it should be turned into an administrative building or an alumni center. A lot of the core intrastructure on campus needs to be replaced and most of all the central laboratory facilities on campus are reaching the end of their lives and they will need to be replaced in the next ten years. We are looking at a comprehensive requirement to rebuild the physical plant in the next ten years. This will take big bucks and it has to be driven by a very clear picture of where we want to be at. There has never been a physical master plan for the campus. The buildings have just come up as we've raised the money for them or borrowed the money. Now we need to step back and see what we want the campus to look like. The 2015 plan should involve this.

Rick asked the President about his scope of vision. Rick sees more emphasis on life long learning. What are the President's thoughts on that? Jack replied the bedrock of what 2015 is

starting from is the idea of an undergraduate focus on math, science and engineering institution on a residential campus. We have to think about a unique value. In his talking with alumni and employers they ask how we need to stay engaged. There clearly is a market for that but that's not what we do. We have a lot of capability going if we choose to go down that path. We may find that that kind of learning is a great source of resources to fund your undergraduate programs. The faculty are contemplating this and trying to decide. Cary mentioned that we tell employers that the CSSE students are life long learners in this department. All our graduates have done life long learning since they have graduated to keep on top of their professions. That's the type of students we get here. We have a firm foundation to build from. Rick mentioned that when he was at Microsoft one-third of the job was keeping up with your education, one third was keeping up with the technology and one third was keeping up with what Microsoft is doing. He mentioned that not enough work force is being generated at the lower levels. The NSF is sponsoring lots of programs to engage students even at the kindergarten level to start to prepare them. There is plenty of money available for this. We do things for secondary education such as Catapult and Homework Hotline. These are great programs. How much of that do you want to do? Who does it? What does this cost? In the competitive market we're looking at Purdue's School of Engineering who is raising $66,000,000 a year; $5,000,000 a month just for the engineering program. They are using that money to upgrade their facilities and to make it possible for their faculty to reach to any student they want. Dennis mentioned that is how he got into Notre Dame by attending a summer camp there. The best indicator as to whether a student is going to come to Rose, predicted by the Admissions Office, is if that student attended Operation Catapult on campus. If the answer is yes there is a very high probability that he/she will subsequently come here. John mentioned that was true when they had Retumpoc here as well from a computer science prospective. Cary mentioned that Rose had a program called Fast Forward geared for girls at the middle school level that lasted four years. John mentioned that he had been contacted by Rose concerning two high school students interested in CS who were considering attending Rose and asking him to write or call them to explain what Rose is. Personal contact might make a difference. Cary mentioned that Erik Jansen, a member of our Board of Trustees, did a cookout for alumni in the Seattle area. Juniors from that area have already contacted Rose about coming here. The CSSE students have started a newsletter that is going to the applicants of the school that have been identified as probable CSSE students. The newsletter goes out once a month with stories about students, faculty and projects in the department. We're willing to do more, whatever it takes.

The school is searching for a new vice president for enrollment management which will oversee the Admissions and Financial Aid offices. Jack mentioned this is a huge step. Most schools now are combining these two rolls. You have one executive responsible for cost, composition, size, quality of your entering class, and four year retention picture. That executive is really the lead on marketing techniques. Chuck Howard is retiring from the Admissions Office after thirty-seven years. Jim Goecker is now Director of Admissions and has been for several years. Dennis spoke of an idea he has with the cash cow side. Raytheon poured a lot of money into Babson College and they have an executive training center for executives to go to for internal corporate training in more of a retreat type setting. He feels we have the resources to offer that setting hear. Jack feels we could do that. Bob asked what the hold is on the MSE Bachelor's which is one degree now. Jack answered that we're just continuing to look at this. Cary said that we want to make sure this fits into Admissions correctly. Our faculty want to make sure that there is

feedback into the undergraduate curriculum with this that it's not going to be a cost burden to the Institution. This is envisioned in the Indianapolis market place but there are concerns and we don't want to push too fast. We want to make sure we do it right. We continue to have discussions. Perhaps this will be there in a couple of years. John agrees with Cary's comments.

The President was thanked for coming and his participation.

**Proposed Master of Software Engineering (MSE) Degree**
Presented by Don Bagert

The Committee to Study the Master of Software Engineering Degree, the CSSE ABET Timeline and Handout of the MSE final report were distributed

Don talked about the presentation made to Alumni in Indianapolis on the evening of February 9, 2005. They did some focus groups with local companies earlier in the day.

Don reviewed the slides from the three page handout. There are a lot of people needed and they need to retrain these people who have been educated in other things. Dennis feels this is a very important point in the software engineering area. Don spoke that we are considering an MSE based in Indianapolis. The audience for this would be working professionals who already have some type of Bachelors degree and would have a background in the core computer science subjects. The idea is that this would be fairly unique in that regular Rose-Hulman faculty would teach virtually most of these courses. Dennis asked why it is important that all CSSE faculty teach this course. Don replied because the Rose faculty have extensive industrial experience. Steven and Salman would teach in this. We wanted to put ourselves apart. IUPUI uses mostly adjuncts. They don't have Ph.D.s and they don't have extensive teaching experience but that would not be ruling these people out but we would be hands on in another city. Cary stated that the graduate programs at Rose-Hulman need to feed into the undergraduate programs as well. If the entire faculty in this program were adjunct faculty we wouldn't be having that dialogue. We would also believe that the school feels it is not quite an important if we don't have the tenure track or tenured people teaching in that program. Dennis commented that the right well chosen adjuncts can be an addition to the program. Cary agreed with that and that there will be adjunct faculty in the program. Don wanted to make sure that primarily it will be regular faculty because the students would be paying big bucks for these classes. Don stated it would be an MBA type of degree. We want to call it the MSE degree to make it clear that we were not in a very vague mission. The mission of the MSE degree is to develop leaders in Software Engineering for quality software education. Dennis feels this degree should be a little more focused. Your target is working professionals that need to be in education, in software education. Yes we want leaders but there is nothing in there that says they are local professionals. Don mentioned the reason for leaders in the previous version was too weak. They want people to bring back to their industry cutting edge ideas. Dennis asked if we assume a CS background. Don stated we assume a small core and that if a person doesn't have it then a person needs to have. Dennis stated we should say working professional. Software professionals is too strong a statement. Cary recommended changing software to working professionals. Jeff mentioned that his employees would not take this degree because they already work in the core. Don stated that this is not intended for those who have a BS in SE but those with a BS in computer science but it

might be useful if they had a very liberal arts background. Our own BS graduates in SE would get less out of this. This is not for people who are steeped in SE. John asked about any work experience and any requirements. Don stated there are requirements. Dennis commented that the purpose of the Master's degree is to achieve some vision. Don responded by stating that the vision is to offer a Master's degree. Cary mentioned that we will need additional faculty to support this program once it has started. Dennis mentioned that there was not a software project management course included. Don stated there was an undergraduate course. Terry Schomacher, Engineering Management, said he would tailor a course into this.
Jeff asked with this being in Indianapolis would it be possible for a student to spend another session in the summer to work on a project for a client like at his company and then go another session in Indianapolis where they are still taking a couple of their core BSSE classes but also be able to work? Don said something could be worked out. John mentioned that this differs from an MBA or law program. Bob supports keeping the project. Rick taught in a program similar to this but there was no project but people brought in ideas. John feels if there was a project he might not consider doing the program. Dennis feels the project should be made an option. Dennis mentioned MSU is very successful in teaching one weekend a month in New York City and Fort Worth. Don mentioned this has been talked about here. Dennis mentioned that companies are more willing for employees to take off one Friday a month. This encourages students to get into these programs. Cary and Don have met with Art Western, Dean of Faculty and President Midgley about this and have kept them in the loop since July 2004. The President expressed concerns about the financial liability. He did this without looking at our data that was just compiled. He suggested an alternative, going to a synchronous distance learning type of set up – collaborative. We would teach to Alumni all over the world from here. We would have distance education classes. He expressed concern about off campus teaching. There was at the time a proposal that the President knew about. When people teach off campus now they only get reimbursed for travel. It was suggested instead for them to receive a small stipend. That idea has died. The combination of these two things has made it so that we feel like a MSE in Indianapolis in not a politically viable thing although we feel it is economically viable and would benefit the Indianapolis community. We intended to have this at the Curriculum Committee in March 2005 and we held off on it and submitted it eventually in April. It has not been discussed by the CSSE Faculty yet because Don was gone but it will be discussed next week. Don's feeling right now is that under the present climate there is no chance that we can have an MSE in Indianapolis approved or if it was approved funded. We also have some staffing problems now. We felt that we would have to hire another person, possibly an adjunct. Don asked if the Board of Advisors could offer any encouragement for this it would be appreciated. Don this is a viable program and that all the CSSE faculty support it.

Cary mentioned the compensation models. Our normal workload for the CS Department is geared around a twelve contact hour a week. It would be a three hour commute from and to Indianapolis. Commute time needs to be figured into the workload. Commute time is part of your contact hours or you get paid additional time you are away from your office. Dennis can substantiate the synchronous teaching with his MSU experience. When this went to video tape or CD within two years no one wanted to set in on this type of class. Cary mentioned that was built into the financial model forecast for this program. The compensation for the travel time could be built into the program so that it would not be a financial burden to the institution. The faculty as we hire will either receive extra pay for their travel time or extra position or part time

position to cover the release time whatever and not effect the upper graduate program at all at Rose and still bring in a positive cash flow from the Master's Degree into the school and Department. Don summarized by saying that we feel like we have a viable program proposal but feel frustrated about the fact of working under one set of assumptions from July 2004 to February 10, 2005. We need to know from the Board of Advisors if we need to go forward or not. Dennis feels one weakness is in putting all one's eggs in one basket in Indianapolis. Dennis mentioned that at MSU they have several options where one can take up the slack. Nate commented about his experience at Rockwell where they have had several programs put together at Iowa State and Iowa University for six or seven years to do synchronous distance learning. It requires a lot more work and commitment from the company. If you can't take time off from work during the day they have everything archived on servers. The down fall he sees is that you do not get the interaction. The strength of our school is in our faculty and in the way the faculty delivers to its students which needs to evolve. We have a low faculty/student ratio. The students know that they are cared about and that their input is valued. Simon asked Don if his report or analysis included what the other alternatives or options were for individuals who might seek this out. Is this a completely untapped market type of thing? Don stated that IUPUI has some courses. They have a Master's in computer science and some courses but there's really not much there. Steve did a thorough search at IUPUI to see if they were planning a Master's degree program in SE and no one knew anything about any thoughts on it. We believe there is not any competition for onsite but there is competition at the National Technological University. Dennis asked if anyone has investigated the cost of synchronous education. Cary is head of the Rose 2015 collaborative which we are hoping to leverage some of this over. Bob asked Dennis if all three of the modes started at the same time or do they develop over time. Dennis responded that prior to 1968 it was all the standard classroom stuff. In 1968 the synchronous learning was started and went for twenty some years as soon as the other options became available. Don mentioned that the focus groups gave some real great information. Some good as well as things we didn't want to hear and if we follow with this input we will be successful. Unfortunately, it's not going to happen in the near future.

**Rose-Hulman 2015: Vision for the Department**
By Cary Laxer

The Summary of the feedback on the CSSE 2015 discussion was distributed.

Cary reviewed the 2015 discussion handout.

Cary asked the Board how they see the CSSE Department in 2015. The Alumni have already received information about this. The Dean and the President have asked each academic department to pitch in from a department focus on this. What does the Computer Science and Software Engineering Department look like in 2015? What should our department be like? Our curricula, students, faculty and equipment in the labs. What's our vision? The CSSE Department put up a forum for our Alumni and asked our current students as well. The handout gives a synopsis of this. The Department faculty met this week and had their discussion about that. The Department brainstormed three hours after that and thanks to the tremendous effort of Lynn to compile this. It is still very much in rough form. Cary had to submit a four page report

to the Dean in mid May that people will ultimately share with the Board of Trustees and all the other 2015 forums and discussions. We also want to get the Board of Advisors input into this. You can do that with or without reading this handout. What kind of input does the Board of Advisors have as to what our Department, our focus, what we should be liked ten years down the road?

Jeff mentioned that even when you are working in the software industry it is incredibly difficult to keep up with changes. Jeff would like to see more opportunities for industry to be able to work or speak with a professor led class with a survey of methodologies and have different people who are in the field to come in and show what they're doing to the undergraduates. Process is what Jeff has to teach the employees he hires. They have different approaches and it's very difficult to keep up to date. Here is what is needed.

1. more opportunities for industry to collaborate with teaching, guest lecturing
2. more and more classic development – integratation of elements, components, standards of web services and services oriented architectures
3. System engineering, make the system work

Dennis feels Rose-Hulman is a magnet, more of a place were things are happening rather that a place you go to hide. Rose-Hulman right now is a deep dark secret in the country side in Indiana, somewhere but no one knows for sure. Whereas for example, Carnegie Mellon is viewed as a center where things are happening. There are things for undergraduate students, graduate students; professional activities going on there, companies go there, they send their executives there for things. Another example is Babson College who is a center for business. They have executive training programs, undergraduate programs and graduate programs but if you are going to study business this college comes to mind. One of the ways Rose could expand their role is to be a place where things are happening in related subjects. For example, you hold an event that world leaders in some things are all coming together to talk about something and people come to hear them. Similar to our www@10 conference we had in the fall 2004. People associate your name with this. They say they will go there to learn. He feels the undergraduates will really benefit. You don't have to take your eye off the ball of the undergraduate program necessarily but you create a much richer environment. Just by tapping Indianapolis you could bring a lot of people from this area into the classrooms and collaborate in the classrooms. Cary mentioned that many offers have been received from many professional people to help and be involved with any opportunities to assist. Cary and Mark have both used some guest lecturers in their classes. David is suggesting a current practices course that the students have asked for where you bring in current practicing professionals to do those current practices. Dennis recommended his company as well as Microsoft could come in to talk to the students. John suggested building a community feel, a sense of community around the program. Lilly uses this idea with their employees by instant messaging where the employees are still very connected and have a very community type of field. Establishing that community feel for the department itself. David asked with whom to have this feel. John recommended within the Rose campus community. Cary mentioned the department has a fall picnic and pool party where everyone shows up and we have a cookout at the end of the year. Dennis mentioned Emory-Riddle made themselves the center of the personal software process. People come to learn about it, the students come to latch on and the day before they have a panel of the visitors asking students

questions. Cary mentioned that we have done that several times but it seems to be dying a little bit. Dennis feels there is a danger in having only an undergraduate program and of it becoming isolated, so self centered that nobody else outside your cocoon knows you exist. You can approach that with some of the things that are on the board. You can still have an undergraduate focus without becoming so isolated. John mentioned about the Solar Car program we had that John heard a lot about in the press or unrelated settings where high school kids were asking about Rose-Hulman. Once the program died all that was seen in the newspapers about Rose was the rallies. Nate commented on Emery-Riddle. They have made the tier one list for recruiting for Rockwell. Several years ago he had not heard of this but they are now on the top list. If you get our name out to more companies to recruit then the parents will hear Rose's name, then the kids will start hearing about Rose also. Nate is trying to get Solar Phantom, aerial robotics and a few others to Rockwell to give presentations to get our name out. Jeff suggested trying to find someone in companies to come to campus and talk to classes. Jeff mentioned that Rose has had a lot of retirements over the last three years. Bill Sisson, Director of Corporate Relations and Placement, and Ron Reeves, Vice President for Development and External Affairs, knew a lot of people in industry. This connection is being lost. Long time recruiting companies that use to come every year to Rose are no longer coming. A lot of the connections we had at companies have also retired.

4. Industry collaboration – design into classes.
5. Recruiting

Rick mentioned that Microsoft has a mission statement for learning through 2020 and they have developed this through the collaboration with the Association of American Scientists Department of Education. A lot of these things are in that vision but a couple not brought up are not. Augmented reality as being part of the education process. John asked what this meant. It means virtual reality. This is tied to real environments remotely with education space. Example, the Hubble telescope. Students actually being able to get the Hubble to do something for class. Robots at the bottom of the ocean. Augmented reality seems to be very promising.

6. Augment reality
7. Collaboration beyond just industry academia – industry government academia It is a level of collaboration which you work together to achieve a common vision. Come up with something you can all work on together. It extends beyond just one educational site. For example, Research Triangle Park. They have special interest groups throughout the park that talk about robotics, embedded systems, process and who cares what university you're with or what company you're with.
8. Role play in teaching using industry reps
9. Big name people on campus – weekends, week days.

David mentioned our WWW@10 conference held in September 2004. It was successful with our students but not with bringing in outsiders. John asked what kind of audience we want to attract. Cary suggested having a weekend where you could attract high school students one day, professionals one day, etc. depending on the focus. John mentioned that Indiana University has a very successful business focus in Indianapolis which attracts a lot of people. Cary mentioned that the CSSE faculty are trying to get out there by going to conferences. Also, Frank Young and

Cary have been successful in making people aware of Rose-Hulman by being the registration co-chairs for the ACM SIGCSE (Special Interest Groups Computer Science Educators) conferences. John suggested you have to get people talking, thinking and hearing it before people will take the next step and study, investigate and really go after it. The challenge has been that we have skipped step one in some ways.

Cary gave an update on where the accreditation process is. When the accreditation process takes place in the fall 2006 we will ask several of the Advisory Board members that are close by to come to campus for several hours to meet with the site visitors. There are two ABET accreditation boards that are of interest to the CSSE Department CAC (Computer Accreditation Commission) which will credit the computer science program and the EAC (Engineering Accreditation Commission) which will accredit the software engineering program. Each one requires separate training to become a program evaluator. We have expertise in both. Frank Young is trained to do evaluations in CAC and EAC, Don is an EAC evaluator, Cary is a CAC evaluator, Dennis is an EAC and CAC evaluator and Bob Cannon is a CAC evaluator. Larry is doing the self-study draft for computer science and Don Bagert is doing the software engineering draft.

**Accreditation Update and Discussion**
Presented by Don Bagert and Larry Merkle

The handout of the ABET Timeline was distributed.

**Software Engineering**

Each program needs to have a program assessment showing how we are going to assess continuous improvement of the program each year. It will be due six weeks before the year begins and that gives us time to look at the report from the previous year and to make the necessary changes. Each term we have to have course objectives for the term completed and course assessment plans on how we're going to meet objectives. At the beginning of the next term course assessments are needed and course reports for the previous term. The report must include how the plan was implemented, what date it was collected and what were the results for each course. Those course objective assessment plans and reports are done each term a course is taught for any course required in the CSSE Department. Dennis asked about our using the word objectives when he was familiar with the word outcomes; course outcomes and program objectives. Bob commented on the EAC terms that SE should address. One addresses the program and speaks to what undergraduates might be expected to have after two or three years out of school. Objective and outcomes are what you are speaking of measuring at the end of each offering. Don asked if the program assessment is really assessing two years later. Dennis stated that the outcomes are typically evaluating what this course is suppose to accomplish, etc. While the objectives are what the program is suppose to accomplish Bob mentioned that the EAC wording is coming much closer to the same terms. By the end of the summer meeting something may come out concerning this. Cary mentioned that several of our required courses are required for both CS and SE. Do we need to prepare two documents? One which uses the term outcomes and one that uses the term objectives? Don and Bob agreed this isn't necessary. Rick mentioned that have to have actual CAC program evaluator training in order to bring the

two into alignment. Bob recommended looking at the site on the web. Don announced the Version II of the workshop will be held in October. Don asked if the assessment report really intended to asses two years out. Dennis stated that it is if you recognize that there is a difference.

The Self-study Part I is due to the Dean February 1, 2005. Part II of the self-study is institutional and contains the course syllabi and resume which isn't due yet.

We want to have Frank Young look at the stuff next moth for a very informal mock review. Tentatively we have an external mock review scheduled in September. We are hoping to have CS and SE doing evaluations of the self-study materials to give us a dry run. The Self-study to ABET is due by June 2006. Dennis recommended this be done in October instead of November because the weather here would be better.

Don is trying to get into a cycle to do before next term to have course assessment plans and course assessment outcomes approved for next term. Then sometime into the term review the reports. Page 2 shows the schedule of when things are due. We are having a hard time getting ourselves on schedule. Rick asked if there were any plans for teaching release time for Don and Larry. Cary responded that Larry gets some release time and Don does also as SE Director.

Todd asked about the draft reports due in November and that we are not getting feedback until August. Dennis explained that there is an extreme process involved including an audit check, quality check, system check, etc. to review the draft and comments. Rick asked if there are any questions that need to be cleared we can get answers to the evaluators before the accreditation is done.

**Computer Science**

Larry showed documents that we developed that we are keeping on a Twiki site. As an example Larry showed the CSSE 332 course objectives. Dennis asked how we know how you've assessed them. Larry responded that this shows up in the course assessment plan. Once all the objectives have been approved for a term we are ready to move onto course assessments. Dennis stated that documenting this makes this helpful. Dennis asked if there is there a mapping to the ABET outcome? Larry reported that this will be done in the higher level of the program assessment plan. Dennis explained how this relates to the ABET. One of the benefits is that it serves as a reminder so that you don't lose track of what you're trying to do. Larry explained the Matrix. Dennis mentioned that this seems rather general and that it could almost apply to anything. Dennis asked how would tell about this Matrix. Larry showed the course assessment report of CSSE332. Dennis only showed the ones that did not. Larry has a statement about the other objectives that were met. Dennis has seen the bar graphs.

Dennis what is purpose of this institute – reason for being? The objectives should embody this somehow. What is the difference between CS and SE? What is the fundamental difference and what will happen to graduates after they leave Rose-Hulman?

Bob in some cases is there relevant student feedback to reflect the recommendations for the next course teaching? Yes this shows up in the changes for the next course offering. Dennis asked

what the criteria are if a student did a poor job would they actually get credit somewhere. Dennis asked where does a prospective student learn where the program objectives are. Larry replied these can be found on our department website. Cary mentioned that the Dean has gotten all the academic departments' websites to look similar. Larry asked Cary if there is a template for the catalog. Is this just something we put in? Larry asked how to find the white paper. Larry will talk to Bob during break about finding the white pages. Bob recommending reading criteria #2 and #3 of the ABET guidelines. Bob suggested following the EAC terminology. Dennis mentioned that these are a lot harder to state and measure. JEFF asked if we were finding this tool limiting. David replied that we were not. Dennis recommending rewording from software engineering majors receive to software engineering graduate will excel. Dennis stated this forces you to keep track of your graduates. Don stated that we will have seven SE graduates this year. It was recommended to get some kind of feedback from them and their employers. Dennis mentioned that he did not notice anything to do with safety in the CSSE labs. Is a student going to do something dangerous, like something electrical? Jeff recommended doing a review of the area and documenting it. Cary mentioned that Mike Howard is our Safety Office on campus. He will have him check the labs.

**Wrap-up Session with Faculty**

John was the Facilitator for the Advisory Board.

The Advisory Board continues to be impressed with the caliber of students and faculty that Rose seems to be able to attract and would like to continue to see this. They also appreciate the opportunity to contribute to the success of Rose. They acknowledge the faculty for their commitment to the students, Dr. Midgley, for explaining the increase competition, shrinking pool of students available and 2015, Lynn for her good minutes and the students.

The Board feels is is critical to focus on core mission. To not lose sight of the core mission of the Institute providing world class CS and SE education through remaining a teaching institution. The Board has concerns about the synchronous learning. They think the collaboration is good but to build on something else but not of itself. The synchrous distance learning would be the worst of both worlds because it is neither personal nor flexible. Classroom instruction is the best and most clearly aligns with the Institute. They think asynchronous distance learning could be a possible supplement but didn't see a major draw to offer some kind of degree online type of offering for Rose.

Cary one again thanked every one for attending and their input.

**ROSE-HULMAN INSTITUTE OF TECHNOLOGY**
**DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING**

# BOARD OF ADVISORS MEETING
# MONDAY, MARCH 13, 2006

## AGENDA
## CSSE CONFERENCE ROOM, MOENCH HALL

| | |
|---|---|
| 9:30 AM | Continental breakfast |
| 9:45 AM | Call to order and introductions |
| 10:00 AM | Bob Bright, Chairman of the Board of Trustees and CEO |
| 10:30 AM | State of the department report by Cary Laxer |
| 10:50 AM | Meet with students in Steve Chenoweth's CSSE 377 (Software Architecture and Design II) class (Olin 157) |
| 12:00 Noon | Lunch, informal discussion (Heritage Room of Union) |
| 1:00 PM | How can the Board help us evaluate emerging technologies that we should include in our curriculum? For example, service-oriented architectures (CORBA, J2EE, .NET) and scripting languages (PHP, PERL, Python) are becoming more popular. What are the industry trends we should consider? Is object-oriented programming really necessary or is imperative programming good enough? |
| 2:30 PM | Break |
| 3:00 PM | Accreditation update and discussion |
| 4:30 PM | Executive session |
| 5:00 PM | Wrap-up session with faculty |
| 5:30 PM | Dinner with some CSSE faculty (Stables) |

**Rose-Hulman Institute of Technology**
**Department of Computer Science and Software Engineering**

**State of the Department Report**
**March 13, 2006**

- Overall, the state of the department continues to be excellent, morale, for the most part, has returned to its usual very high level following last year's presidential situation, and the outlook continues to be great.

- The department faculty and staff continue to operate very well together. Everyone is a team player and puts the needs of the department and our students above individual concerns. Three wonderful new colleagues joined the department this year, and they have been well received by our students:

    - Matt Boutell - Matt joined us from the University of Rochester, where he completed his Ph.D. in computer science. Matt's other degrees are an M.S. in computer science (also from the University of Rochester), an M.Ed. in mathematics education (University of Massachusetts), and a B.S. in mathematical science (Worcester Polytechnic Institute). He has experience teaching at both the high school and college level, and has been elected to *Who's Who Among America's Teachers* three times. Matt's research interests are in computer vision and machine learning, and he has had close collaborations with Eastman Kodak Company. Matt and his wife, Leah, have four children: Jonathan (7), Caleb (5), Elise (3), and Elliot (1). Besides spending time with his family, Matt likes to play the tuba and hike.

    - Curt Clifton - Curt received his Ph.D. in computer science from Iowa State University. He also has an M.S. in computer science and a B.S.E.E., both from Iowa State as well. Curt has industrial experience in manufacturing management with Procter and Gamble, held positions as a software team leader and a senior electrical engineer for L&S Electric, and served as an independent consultant. His research interests are in the areas of programming languages and aspect-oriented programming. Curt is married to Lisa Laxson, a physician specializing in family practice and obstetrics. They have a 11-year old beagle, Molly. Curt and Lisa grew up in Earlville, Iowa, just a few miles from the Field of Dreams. Curt enjoys woodworking and bicycling, and he and Lisa both enjoy traveling, hiking, and camping. They are big Chicago Cubs fans.

    - Lisa Kaczmarczyk - Lisa finished her Ph.D., an interdisciplinary one with a focus on cognitive science, this past summer at the University of Texas at Austin. She holds two M.S. degrees, one in computer science from the University of Oregon and one in information systems from Northeastern University, and a B.A. degree in Spanish and Drama from Tufts University. Lisa has many years of teaching and curriculum development experience at

the college level, and has served as a design consultant and software engineer in industry. Her research interests lie in the area of cognitive science. Lisa is fluent in Spanish. She has a cat named Goddess Maya Pinecone ("Maya" for short), and likes hiking, camping, mountain climbing, coffee shops, used book stores, travel, and museums.

- Thanks to the support of the Dean, we now have 13 full-time faculty in the department, which enables us to support two people on sabbatical every year without having to hire adjunct or visiting faculty. This year Claude Anderson and J.P. Mellor are on sabbatical. Claude is working at ANGEL Learning Labs in Indianapolis. They are the developers of the ANGEL courseware management system that we are using. J.P. has been teaching at California State University, San Luis Obispo. Next year Mark Ardis and Mike Wollowski will be on sabbatical. Mark will go to the Software Engineering Institute at Carnegie Mellon University, and Mike is going to Germany to do research on smart dust, an application of swarm intelligence, at the University of Luebeck.

- We continue to try to provide our students with the facilities they want in our labs. This year we used our share of the institute's equipment funds to acquire 12 docking stations, flat panel monitors, keyboards, and mice, for students to plug their laptops into. Our students really love this approach to the equipment – it let's them use their computers with a bigger screen and keyboard, facilitating team project work. We are also in the process of evaluating electronic whiteboards for use in our labs, which will enable students to do work on the whiteboards and then download what they have done onto their computers. We hope to have at least one electronic whiteboard in each of our labs by the end of the year.

- Our new software engineering major has been well received. Winter quarter enrollment data shows 58 software engineering majors in the department, along with 138 computer science majors. Most of the courses are settling down nicely, although the two that are required for both computer science and software engineering majors receive comments from students that they don't see the real need for these courses (they will a couple of years after their degrees!).

- The master's degree in software engineering is on hold. The school's Board of Trustees is taking a good look at the graduate education at Rose in general, and with the turnover at the presidential level at the school we have not moved forward lately until we are sure we will have the support needed to make the program successful.

- This year we are actively pursuing accreditation for both our computer science and our software engineering degree programs. Both will be considered for first time accreditation when ABET visits the campus in the fall for its next general review of the engineering programs at Rose. Our computer science program has been chosen to be one of five pilot programs to be reviewed under proposed new accreditation criteria for computing programs that will make computing accreditation look more like engineering accreditation. Larry Merkle is taking the lead on writing the computer science self-study, and Don Bagert is doing the software engineering one. Tomorrow

Bob Cannon will be doing a mock evaluation of the computer science program for us, to help us prepare for the actual one in the fall. Dennis Frailey did a similar thing for our software engineering program last fall.

- Our collaborative efforts with Uppsala University in Sweden continue. This year four students worked with their counterparts at Uppsala on the Swedish pilot of a European Union-wide project called SPEX – SPreading EXcellence in healthcare. The project looked at ways technology could be used to improve healthcare at more rural locations, bringing the expertise of larger city doctors to the rural locations via technology. Two students traveled to Sweden in December to be part of the project presentation at a workshop and to the client. In addition, two students traveled to Barcelona, Spain in January to be present for the final SPEX conference, where representatives of the three pilot projects (Sweden, Spain, and Italy) came together to discuss the project as a whole. Our students were well received, and were the ones who demonstrated some of the technology to the Deputy Director of the European Health Management Association.

- We are actively looking for other ways to increase the global experiences of our students and faculty. In January I visited the University of Aizu in Japan, the Chinese University of Hong Kong, and Deakin University in Melbourne, Australia, to feel out opportunities for our students and faculty. We discussed some joint, short, technical courses, student projects, student exchanges, and faculty exchanges. All three schools were receptive, and dialogue is continuing with Aizu and Deakin at this time. I am hoping that CSSE can be the standard setter for global educational opportunities at Rose-Hulman.

- Professionally, the department faculty are very active. Every faculty member has attended at least one professional conference in the past year, and many have attended more. Our faculty are increasingly involved with leading the conferences, with several appointed to conference committees.

- Service learning is an important part of what we do. Several senior projects are for community service organizations. Thirty-two students are taking a Teamwork and Robotics course this term, where service learning projects will play an active role in the course. Our students are teaching Botball teams from middle schools and high schools how to work as teams and how to do the robotics work. The course also lets our students learn advanced teamwork methods which can benefit them in industry.

- We have an active group of female students known as gEECS (girls in Electrical Engineering and Computer Science). They are doing wonderful work, including tons of outreach, working with girls at Woodrow Wilson Middle School, working with girls that visited Rose-Hulman, winning the programming award last year, and more.

- Last month eight of our female students and both of our female faculty members attended InWIC 2006, the Indiana Women in Computing Conference. This is a conference to build support for women computer science students around the state. Our own Lisa Kaczmarczyk was invited to give the keynote speech at the conference, and she gave the most wonderful, uplifting, and supportive speech

imaginable for the audience (over 130 women). The conference organizers had nothing but the highest praise for Lisa after the conference, telling me it was the best keynote speech that could have been given.

- Three Rose-Hulman students, Michael Kuehl (CS major), Brandon Cannaday (CS), and Charlie Key (computer engineering), worked on a project with Touch Technologies, Inc., a software research and development company based out of San Diego, CA. TTI offers high-level consulting and product sales worldwide and TTI's high-end software is currently in use by the majority of today's Fortune 1000 companies. The people there were concerned about working with undergraduates and rather hoped to work with graduate students. They were also concerned about our students inadvertently making public their proprietary software. We assured them that they did not have to worry about either and they were very happy with our students.

- Two Rose-Hulman CSSE students, Michael Lehenbauer and Tyler Hicks-Wright, were part of a team of four summer interns (out of 800 applicants) at Fog Creek Software in New York City last summer. A documentary of their internship was made, and the film is now available on DVD. The title of the film is “Aardvark’d: 12 Weeks with Geeks,” and we have shown it on campus. Our two students are prominently featured, and there are shots of campus in the film.

- CSSE students are prominent among those experimenting with the use of Tablet PCs and DyKnow. David Mutchler, a CSSE faculty member, is on the team that won a recent $100K award from Microsoft to assess that use.

- Our graduates are in high demand by industry and leading graduate schools. 2005 graduates were employed by Xetron, Expedia, Microsoft, ITT, Guidant, Software Engineering Professionals, Interactive Intelligence, ANGEL Learning Labs, Rockwell Collins, Beckman Coulter, United Space Alliance, and Webfoot Technologies. Graduate schools that our 2005 alumni have been admitted to include Indiana University, the University of Pennsylvania, and Stanford University.

- The following unsolicited testimonial that I received from a student that graduated last month sums it all up: “*I just wanted to say that I am very thankful for all that you have done for me while I have been your student and your advisee. I was able to brag about how you treated me at SIGGRAPH to students at schools like Carnegie Mellon and MIT and they said that that would never happen at their schools. Because of efforts like yours, it is no wonder Rose-Hulman is the best school in the universe. Thanks for helping to make my experience here the best undergrad experience I possibly could have had anywhere.*”

- In summary, things are really good for computer science and software engineering at Rose. We have a wonderful and dedicated group of faculty and staff that believe in the mission of our school. Thank you for your support and guidance in helping us achieve what we have so far, and we look forward to your continuing involvement with us as we continue on our mission.

**CSSE Advisory Board Brainstorming**
**3/13/06**

The Board was asked to brainstorm the question: "How can the CSSE department or the Board help improve our program so that students get exposed to appropriate new technologies?"

The suggestions were grouped into 6 categories:

A. **Offers**: things that Board members can do
B. **Open Source**: student involvement with open source projects
C. **Partnering**: ways that students can work with others
D. **Outside Input**: getting advice and information from others
E. **Let It Happen**: ways that students can be exposed to new technologies
F. **What We Think**: general advice

## A. Offers

1. Night before panel [a panel of Board members discussing careers, technology, etc.]
2. SEP guest lectures
3. Clearinghouse [need to provide a clearinghouse of needs and offers]
4. Community oriented projects [some Board members have access to projects that would be good for students]
5. Guest lectures on other careers

## B. Open Source

6. Leverage open source projects to get “real world”/constraint-type experience (i.e. live with what you build)
7. Get involved with open source projects
8. Work with open source initiatives that use emerging technologies (i.e. Mono for NET)

## C. Partnering

9. Formalize relationship between CS Department and Ventures to be intentional about what tech is learned or perhaps use Venture projects in classes to teach or demonstrate a technology
10. Find more opportunities to market the experience gained through Ventures to CS students
11. Explore opportunities for students to take technical training classes from vendors
12. Trend to consider – Netcetric operations, IT a partner as opposed to just infrastructure
13. Of course, the classic “guest lecturer” from industry
14. Increase visibility of internship opportunities

**D. Outside Input**

15. Could do employer survey – in some ways picking one of two techs and learning it better than none
16. Industry needs – Breadth-first is foremost – good companies look for student hires for life – not just current hot languages or trends
17. Consider impact of regulatory environment on development practices. What is different about developing a medical device, for example?
18. Bring in more outside industry speakers to talk about tech
19. Poll alumni for ideas of new technologies that they would want new hires to know
20. .NET technology is critical
21. Trend to consider – Follow ISO adoptions - .NET CLR is moving from corporate defacto standard to ISO – SUSH [???] indicates long term trends
22. Evaluate emerging technologies – Board can advise on corporate trends and thrusts (as allowed)
23. Industrial evaluations at end of projects

**E. Let It Happen**

24. More projects where students pick technology
25. Student tech surveys where they research and then present a technology
26. Give students a chance to choose their own technologies in courses (i.e. which DB to use in DB class)
27. Give students a chance to work across classes – i.e. seniors managing juniors

**F. What We Think**

28. SOA is all I hear about now. Need to develop this expertise at Rose-Hulman
29. OO is well established and is now foundational
30. Teach principles, not technologies. Corba, J2EE – who cares? Both work.
31. Structure and methodology is important but is there an opportunity for more RAD-type experience (i.e., entrepreneurship internships, etc).
32. Trends – oo is really necessary – add experience with architecture generators
33. Provide more programming experience for upper class students
34. Trends to consider – information security and assurance are vital – students need to have a culture of design – in security versus bolt-on security
35. Give student exposure to all the technology types – service oriented, scripting, procedural, oo, etc. Not every implementation, of course, but at least one example of each.
36. Trends – Common sense is essential in SE - CMM/TSP fit some needs, agile fits others

# Rose-Hulman Institute of Technology

## Department of Computer Science and Software Engineering

## CS/SE Board of Advisors

| | | |
|---|---|---|
| To: | Dr. Cary Laxer | Department Head for Computer Science and Software Engineering |
| CC: | Mr. Robert E. Bright | Chairman of the Board of Trustees |
| | Dr. Arthur Western | Vice-President for Academic Affairs and Dean of Faculty |
| Date: | March 13, 2006 | |
| Re: | CS/SE Board of Advisors Meeting Report | |

---

### Meeting

The Computer Science and Software Engineering Board of Advisors met on Monday, March 13, 2006 on the campus of Rose-Hulman Institute of Technology in Terre Haute, Indiana.

Together with members of the CS/SE faculty, the Board members present were:

| | | |
|---|---|---|
| • Robert Cannon | Carnegie Mellon Software Engineering Institute | |
| • Richard Conn | Microsoft Corporation (Retired) | [RHIT Class of 1976] |
| • Bryan Everly | PeopleStrategy | |
| • Skip Franklin | Crowe Chizek and Company LLC | [RHIT Class of 1998] |
| • Jeffrey Gilbert | Software Engineering Professionals, Inc. | [RHIT Class of 1985] |
| • Carl Murray | Beckman Coulter | |
| • John Simon | Eli Lilly and Company | [RHIT Class of 1989] |

This report summarizes the observations and recommendations of the Board. Detailed minutes were captured separately.

### Acknowledgements

The Board wishes to specifically acknowledge the efforts of the following individuals:

- Dr. Cary Laxer and his team of CS/SE faculty for their hospitality, dedication to their students and the Institute and commitment to making an already exceptional program even better
- Chairman Robert Bright for taking time out of his busy schedule to meet with the Board and share his thoughts on the opportunities and challenges facing the Institute
- Lynn Degler for coordinating the logistics for the Board and capturing the meeting minutes
- The students from CS/SE 377 for openly sharing their views and suggestions with the Board

## Department and Facilities

The Board met privately with a CS/SE 377 class, mostly third-year students. The students spoke very highly of the department, many stating that their interaction with the CS faculty and staff was instrumental in their decisions to come to Rose and chose CS/SE degree programs. Dr. Laxer's leadership at the departmental level and Chairman Bright's leadership at the institutional level were both specifically mentioned by students as being much appreciated.

Both student feedback and the State of the Department report show excellent progress in the state of the department's facilities and budget, compared to previous years. Students appreciate the facilities available to them, and were pleased to be involved in the decision process this year that determined the purchase of new hardware (docking stations for student laptops). Faculty members have been able to attend conferences and take sabbaticals, showing excellent improvement in the number of available faculty and resources available for their professional development. The Board hopes to continue to see improvement and growth in this area in the future.

## Student Feedback

Some concern was expressed by students about the results of the feedback they give to the department. Students felt that they were left out of the loop with respect to some feedback they give, such as the discussions with the Board and end-of-term class evaluations. While the students understand that they may not be privy to every result, they would like to be kept informed of the actions being taken based on their feedback where it is appropriate.

Another source of student feedback that has been neglected recently is the Student Advisory Group (SAG). Students indicated that this group has not met at all this year. Since this group is an important forum for student involvement, as well as being mentioned by name in the ABET accreditation process, the Board feels that the CS/SE department should make a concerted effort to make the SAG an active group. While students, especially incoming freshmen, may not immediately understand the importance of the SAG, the faculty do understand and should be responsible for encouraging the students in this important effort.

## Software Engineering Bachelor's Degree Program

Students expressed some concerns over the Software Engineering program. They felt that there is not enough choice in the SE course structure – all the SE core classes are required, with no elective options. The students realized that this is something that is difficult for a young degree program, but wanted to encourage further course offerings in the future.

Students also felt that the higher-level courses in Software Engineering did not require enough coding. Some students were able to move through their last two years with little or no responsibility for actual coding assignments.

The Board discussed these issues with the faculty. We feel that the issue of electives will be naturally resolved as the Software Engineering program continues to mature. The coding issue is very important in our view, since nearly all new CS/SE graduates that move into the corporate world will be expected to be excellent coders for their first few years. While some of this may be student perception – we were pleased to hear that CS/SE 375 requires individual coding efforts – there may be room for improvement in the way that student teams are determined and graded that would encourage everyone in the team to take responsibility for the coding.

## Student Exposure to Industry

The faculty asked that the Board consider some ideas for areas of improvement in preparing students for real-world software engineering jobs. Many suggestions were offered during a brainstorming session. A few of the most commonly heard suggestions:

- Use of Open Source projects to teach cooperative development
- Allowing students to choose technologies that they use in class (i.e. which database to use in a database project)
- Provide overview of common frameworks in use today, such as Java's Spring for web development or Microsoft .NET
- More guest lecture opportunities to give students some exposure to current industry trends

With respect to the guest lecture suggestion, the Board agrees that this is an excellent opportunity for students to learn. Many of the Board members are willing to be involved and have participated in the past. However, in order for such a program to succeed, there is a need for coordination. We encourage the department to appoint a singe person to be responsible for coordinating this effort. There are many people who would be willing to give students a glimpse of how computer science and software engineering is used in their everyday lives – all that is needed is leadership to connect those people and the students.

## ABET Accreditation

The Board reviewed the preparations underway by the department for ABET accreditation of both the computer science and software engineering academic programs. It is clear that much effort and thoughtful consideration is being given to this important process, including mock examinations by Dennis Frailey and Robert Cannon from the Board. In addition, by closely examining itself and its programs, the department has an opportunity to uncover new areas for continuous improvement.

ABET accreditation will even further solidify the already outstanding reputation of the department and Institute as a leader in providing the very best education to the next generation of scientists and engineers. The Board enthusiastically supports continued focus on this endeavor and stands by to offer the department its full support and any assistance it can provide.

## President Search

Chairman Bright updated the Board on the search for a new Institute President. We are pleased to hear that a decision has nearly been reached. Chairman Bright has done an excellent job in the difficult transition period in the last few months, as both student and faculty comments have shown.

We encourage the new President to focus on the fund-raising responsibilities that are so important to the Institute's financial future, and also on building a rapport with the students and faculty on campus. Both are essential for a successful Presidency.

*Note: Since the board meeting, the Rose-Hulman Board of Trustees has announced that Dr. Gerald S. Jakubowski has been selected as the 13th President of the Institute.*

## Coordination with Rose-Hulman Ventures

Some students expressed disappointment that the CS/SE Department has not done more in cooperation with Rose-Hulman Ventures recently. Many students have worked with Ventures as interns, and felt that the department faculty had not been supportive of the effort.

The Board discussed this with the faculty, and found that they were aware of the Ventures opportunities, and had passed that information along to students. However, in doing so the faculty also expressed concerns to the students that the Ventures project management style was not good preparation for their future. The Ventures programs are very focused on end-results in minimal time, and as such ignore many of the fundamentals of software engineering (such as good requirements gathering and adequate test planning).

Faculty indicated that the Ventures management had not been amenable to dialogue about this issue. The Board feels that the faculty has acted appropriately in this situation, and should continue to inform students of their concerns with respect to Ventures. The decision of whether to become involved with Ventures is ultimately left to the student, as it should be.

## Conclusion

The members of the Board continue to be thoroughly impressed with the caliber of Rose-Hulman students and faculty. We very much appreciate the opportunity to contribute to the further success of the CS/SE department and look forward to continuing our efforts at our next gathering. Thank you.