

# PROGRAMMING LANGUAGE PARADIGMS

Curt Clifton

Rose-Hulman Institute of Technology

# TEAM EVALUATIONS

- On ANGEL
- Available after 3pm Friday
- Due by 8am Monday
- Failure to complete will result in a full letter grade deduction for your team project score

# COURSE OUTCOMES

- Write programs in a variety of programming language paradigms using the features unique to those paradigms.
- Identify the semantic similarities and differences between a new programming language and previously known languages.

# COURSE OUTCOMES

- Clearly explain the key semantic features of a new programming language, including how to begin using it and its strengths and weakness.
- Develop medium-sized applications in a new programming language through independent study.

# IF YOU WERE DESIGNING A LANGUAGE...

Quotes from <http://james-iry.blogspot.com/2009/05/brief-incomplete-and-mostly-wrong.html>

# APPLICATION DOMAIN?

1958 John McCarthy and Paul Graham invent LISP. Due to high costs caused by a post-war depletion of the strategic parentheses reserve LISP never becomes popular[1]. In spite of its lack of popularity, LISP (now "Lisp" or sometimes "Arc") remains an influential language in "key algorithmic techniques such as recursion and condensation"[2].

# STATIC OR DYNAMIC TYPES? OR NO TYPE SAFETY AT ALL?

1964 John Kemeny and Thomas Kurtz create BASIC, an unstructured programming language for non-computer scientists.

1965 Kemeny and Kurtz go to 1964.

# OBJECTS? CLASSES? PROTOTYPES?

1970 Guy Steele and Gerald Sussman create Scheme. Their work leads to a series of "Lambda the Ultimate" papers culminating in "Lambda the Ultimate Kitchen Utensil." This paper becomes the basis for a long running, but ultimately unsuccessful run of late night infomercials. Lambdas are relegated to relative obscurity until Java makes them popular by not having them.

# METHODS? FUNCTIONS? PATTERN MATCHING?

1970 Niklaus Wirth creates Pascal, a procedural language. Critics immediately denounce Pascal because it uses " $x := x + y$ " syntax instead of the more familiar C-like " $x = x + y$ ". This criticism happens in spite of the fact that C has not yet been invented.

# CLOSURES?

1972 Dennis Ritchie invents a powerful gun that shoots both forward and backward simultaneously. Not satisfied with the number of deaths and permanent maimings from that invention he invents C and Unix.

# CHANNELS?

1990 A committee formed by Simon Peyton-Jones, Paul Hudak, Philip Wadler, Ashton Kutcher, and People for the Ethical Treatment of Animals creates Haskell, a pure, non-strict, functional language. Haskell gets some resistance due to the complexity of using monads to control side effects. Wadler tries to appease critics by explaining that "a monad is a monoid in the category of endofunctors, what's the problem?"

# WHAT WOULD YOU LEAVE OUT?

1991 Dutch programmer Guido van Rossum travels to Argentina for a mysterious operation. He returns with a large cranial scar, invents Python, is declared Dictator for Life by legions of followers, and announces to the world that "There Is Only One Way to Do It." Poland becomes nervous.

# **Course Evaluations**

**Curt Clifton**

**Rose-Hulman Institute of Technology**

**YOU SHOULD START GIVING OUT 'E'S  
SO I CAN SPELL FACADE OR DEFACED.**

IF YOU DON'T TURN IN  
AT LEAST ONE HOMEWORK  
ASSIGNMENT, YOU'LL  
FAIL THIS CLASS.

YEAH. BUT IF I CAN FAIL  
THIS CLASS, THE GRADES  
ON MY REPORT CARD WILL  
BE IN ALPHABETICAL ORDER!



**CARTOON OF THE DAY**

**PRIORITIES**

# **How are Course Evaluations Used?**

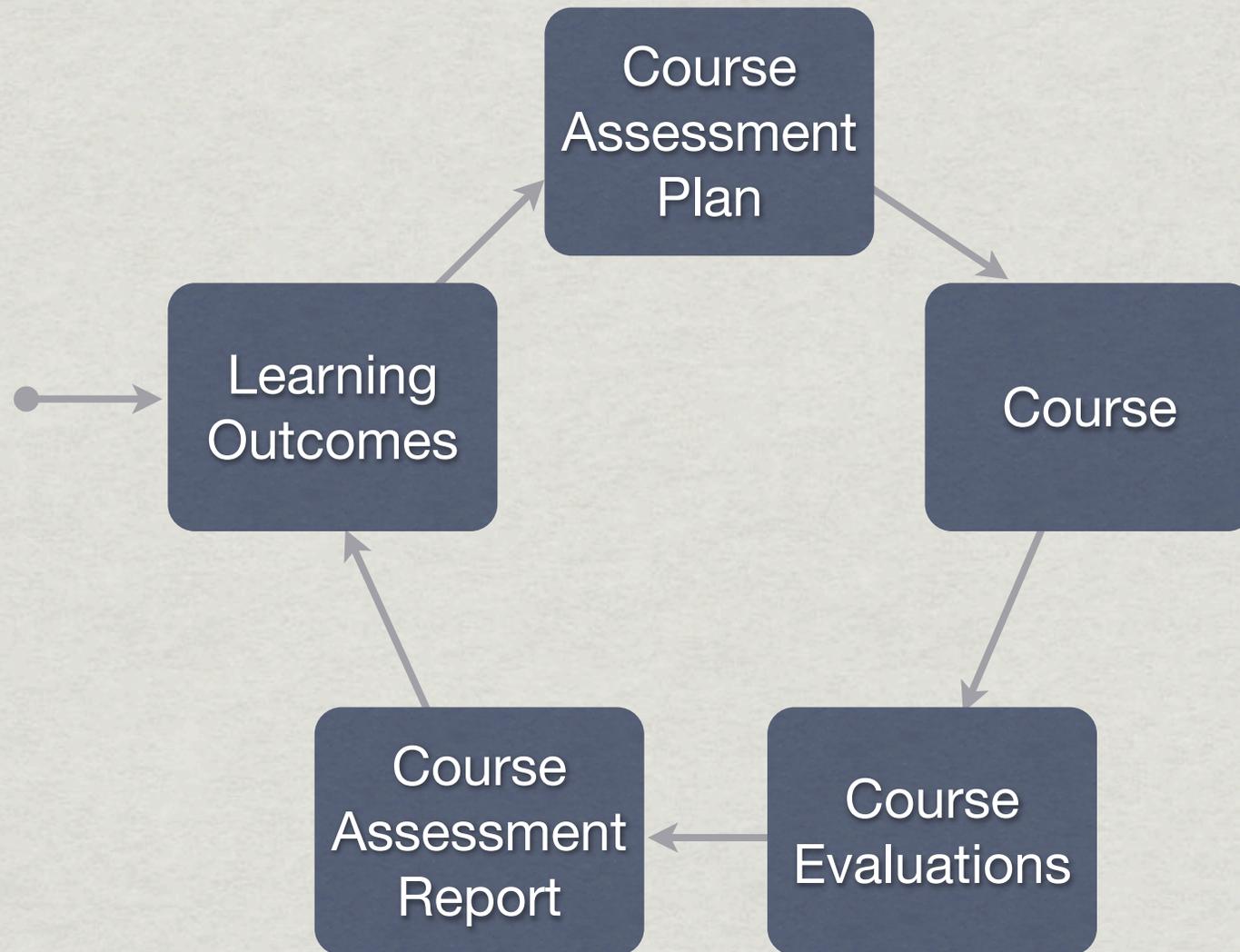
# To Improve Instructor Performance

- \* Comments and summary to instructor
- \* Instructor adds response:
  - \* Plan for improvement
  - \* Explanation for bizarre comments
- \* Package goes to department head for review

# For Promotion and Tenure Decisions

- \* Full set of course evaluations (plus a couple hundred pages of supporting information) goes to Dean and PTR committee
- \* Dean and PTR committee make separate recommendations to President
- \* President has final decision on promotion and tenure

# To Improve Courses and Curriculum



# How You Can Be

# Most Helpful

TO ME, TO ROSE-HULMAN, TO  
FUTURE STUDENTS, ...

- \* Give very specific feedback
  - \* What worked well
  - \* What didn't work, and how that could be fixed
- \* Consider your audience
  - \* Instructor (primarily), department head, Dean, PTR committee



# CARTOON OF THE DAY

A-MINUS-MINUS

YOU CAN DO THIS ONE IN EVERY 30 TIMES AND STILL HAVE 97% POSITIVE FEEDBACK