CSSE 132 – Introduction to Computer Systems
Rose-Hulman Institute of Technology
Computer Science and Software Engineering Department

Final Exam Review Guide

This exam measures your mastery of these learning objectives:

Objective 1 Describe the functions of common computer system hardware elements including CPU, memory hierarchy and input/output devices.

Objective 2 Implement and analyze software in the C programming langauge using:

- Standard C data types
- Binary arithmetic, boolean and logical operations
- Functions
- Arrays
- C Strings
- Pointers and Pointer Arithmetic
- Static and Dynamic memory allocation techniques
- User and file input/output
- Command-Line arguments

Objective 3 Use logic gates and a hardware design language like Verilog to implement standard computational structures including memory cells, adders, ALUs, and multiplexers.

Objective 4 Discuss why certain abilities such as information representation, network communication, input/output, and security require support from multiple layers of a computer system.

Objective 5 Demonstrate ability to perform tasks like these in a variety of operating environments including the Linux system environment:

- compile software
- debug software
- secure files
- leverage a version control system
- manipulate data
- command-line (shell) navigation and manipulation

1 Topics to study

- Numbers (Objectives 2 and 4)
  - Number representation in binary, hexadecimal, and decimal.
  - Conversions from one number system (binary, hex, decimal) to another.
  - Two's complement
  - Operations on binary numbers (bitwise operations, addition, subtraction)
- Logic Design (Objectives 1 and 3)
  - Truth tables and boolean logic.
  - Combinational logic expressions
  - Sum of products form and creating combination logic expressions from truth tables.
  - Multiplexers and decoders
  - Creating circuit diagrams from combinational logic.
  - Creating an ALU that adds and does other operations.
- Sequential logic (Objectives 1 and 3)
  - Basic Memory Cells
  - Latches and Flip Flops
  - Clocks
  - Building a register file
  - Memory addresses
- Memory Hierarchy (Objectives 1 and 4)
  - Different types of memory (DRAM, SRAM, SSD/Flash, Hard Disk)
  - Disk access time
  - Effective access time (given multiple types of memory and a cache strategy and hit rate)
  - When would you use various types of memory?

- Reading ARM Assembly (Objectives 1 and 4) Given a program, what does it do?
- Reading C code (Objective 2 and 4) What does this code do?
- C Programming (Objective 2)
  - Pointers and Arrays in C
  - Functions in C (and procedure calls in ARM assembly)
  - Representing data in a binary computer
    - C data types `float, int, char`, etc.
    - Representing numbers using bits – integers, floating point, fractions.
    - Range and precision
    - C arrays and strings
    - C structs
- Memory Allocation (Objectives 2 and 4)
  - Allocating and using memory on the stack (in C and ARM assembly)
  - Allocating, using, and freeing memory on the heap (in C)
  - Creating a C program from scratch
- Input/Output (Objectives 1 and 2)
  - Similarities and Differences between Buffered and Unbuffered IO
  - Getting input from a user in C
  - Reading and writing files in C
- Command Line operations (Objective 2 and 5)
  - Using SVN to check out, commit, and add files to a repo
  - Editing files on the command line (using `nano, vim, emacs` or another text editor.
  - Examining text files using linux commands like `cat,less,grep,find`
  - Compiling and running C programs
  - Debugging C programs
- Network Programming and Sockets (Objective 4)
  - Basic network building blocks (hubs, switches, routers, etc)
  - Addressing computers ("hosts") on the network
  - The roles of clients and servers
  - Creating sockets on the client and server