

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

Exam 2 Review Guide

This exam measures your mastery of these learning objectives:

- Objective 1** Discuss why certain abilities such as information representation and input/output require support from multiple layers of a computer system.
- Objective 2** Describe functions of common computer system hardware elements including CPU and memory hierarchy and relate “program execution” to activation of various elements using a subset of the ARM assembly language.
- Objective 3** Implement and analyze software in C using:
- (a) Standard C data types
 - (b) Binary arithmetic, boolean, and logical operations
 - (c) functions
 - (d) arrays
 - (e) C Strings
 - (f) Pointers and Pointer Arithmetic
 - (g) Static and Dynamic memory allocation techniques
- Objective 4** Demonstrate ability to perform tasks in a variety of operating environments, including the Linux system environment.
- (a) Compile and debug software
 - (b) use a version control system
 - (c) manipulate data
 - (d) command-line (shell) navigation and manipulation

1 Topics to study

- Reading ARM Assembly (Given a program, what does it do?)
- Reading C code (what does this code do?)
- 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
- 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
- Command Line operations
 - Using Git to clone, pull, commit, push, 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