CptS 121 - Program Design and Development

Final Exam Review Guide

This document will serve as a guide to help you prepare for the final written exam in CptS 121. You will find information about the exam format and topics you are expected to review within this guide.

What to Bring?

- Your WSU ID
- Two sharp pencils
- A "cheat sheet" (see below)
- Calculators and other notes may **not** be used during the exam!

The "Cheat Sheet"

The exam will be closed-book, but you will be allowed a "cheat sheet": **one side** of a page whose dimensions may not exceed 8-1/2" by 11" (i.e., standard sheet of notebook paper). This is twice the size of the "cheat sheets" allowed for the midterm exams. You must present your cheat sheet to your instructor at check-in, so that he can verify that it meets regulations. If you use a cheat sheet that exceeds the allowable dimensions, or that has writing on both sides of the page, you run the risk of its being confiscated prior to the exam. This policy will be strictly enforced.

Exam Timeframe

The final exam is scheduled for:

- **Lecture section 1: Thurs, Dec 12, 10:30 am – 12:30 pm in ABEL 201**
- **Lecture section 2: Tues, Dec 10, 8:00 – 10:00 am in SPRK G45**

Exam Format

Expect the final exam to look a lot like the two midterms, except that it will be longer, because **you will have a full two hours** to take the exam, rather than one hour. There will be a mix of true-false, fill-in-the-blank, and short answer questions that test your knowledge of key concepts. Expect also to supply short code fragments, and to trace through C code fragments and specify their output.

Exam Coverage

The exam is comprehensive, covering all the material we have explored in this course. However, the emphasis will be on the final five weeks of the course (chapters 8 - 10, 12, 13, Appendix C, and pointers of the Hanly and Koffman text).

Topics that are fair game from midterm #1:

- See the midterm #1 exam review for a list of topics that are fair game. In other words, all of the material covered in the first five weeks of the course is fair game.

Topics that are fair game from midterm #2:

- See the midterm #2 exam review for a list of topics that are fair game. In other words, all of the material covered in the second five weeks of the course is fair game.

The following is a list of exam topics covered in the final five weeks of the course. These are the topics that will be emphasized on the final exam, so you would do well to devote the majority of your study to these topics:

Chapter 9: Recursion

- Define what is *recursion*
- Define what is a *base case* and what is a *recursive step*
- Define what is a *recursive function*
- Define what is a function call *stack*
- Diagram the function and return *sequence* of a particular recursive solution
- Convert between iterative and recursive solutions and vice versa
- Apply and implement recursive functions to solve a problem

Appendix C: Bit Manipulation

- Define what is a *bit*
  - Recall a bit is a single binary digit – 0 or 1
- Define what is a *nibble* and *byte*
  - A nibble is a sequence of 4 bits and a byte is a sequence of 8 bits

- Convert between binary and decimal numbers and vice versa
- How can we determine if a number is even or odd without using the mod (%) operator?
  - If the least significant bit (lsb) is 1 the number is odd, if its 0 the number is even
- How do we multiply and divide by powers of 2 without using the multiplication (*) and division (/) operators?
  - Recall multiplication and division is expensive and resource intensive
  - Shift all bits in a number to the *left* by 1 to multiply by 2
  - Shift all bits in a number to the *right* by 1 to divide by 2
- Apply bitwise operations to decimal numbers
  - These include: one's complement or negation (~), left shift (<<), right shift (>>), AND (&), OR (|), and XOR (^)

Chapter 12: Programming in the Large

- Define what is a command line argument

- Modify main ( ) to accept command line arguments
  - The arguments are *argc* and *argv*
- Extract strings from the command line arguments
- Convert strings to integers using atoi ( )

Chapter 13: Dynamic Data Structures

- Define what is dynamic memory
  - Memory allocated at runtime from a memory area called the *heap* or *memory store*
- Apply malloc ( ) to allocate memory
- Apply free ( ) to de-allocate memory
- Apply sizeof ( )
  - Recall this returns the number of bytes allocated for a type or variable

Recommended Strategy for Preparing for the Exam

I recommend that you use the following activities and materials to prepare for the exam:

- **Review quizzes and lab exercises**: These may well be your best resource. An excellent learning activity would be to retake the quizzes and review the lab exercises.
- **Lecture slides and example code**: Study the lecture slides and example code. Continue to complete extra coding examples on your own time.
- **Read the textbook:** Read or re-read chapters 1 -10, 12, 13, & Appendix C in your textbook. Solve the end-of-chapter exercises.