Midterm Exam #1 Review

CS 2308 :: Spring 2016
Molly O'Neil
Midterm Exam #1

• Wednesday, February 24
• In class, pencil & paper exam
• Closed book, closed notes, no cell phones or calculators, clean desk
• 20% of your final grade
• 80 minutes to complete the exam
• Please bring a pencil and eraser!
• All writing will be done on exam paper that I'll hand out
Exam Format

• 100 points total
• ~19 problems, ~8 pages

Types of questions:

• ~60% of total points
  • Multiple choice
  • Fill-in-the-blank concepts
  • Short answer
  • Demonstrate search/sort
  • Tracing pointer code

• ~40% of total points
  • Writing functions/code (similar to coding assignments)
  • Some of this is fill-in-the-blank, some from-scratch
Content

- Slides from lecture:
  - Linux
  - CS 1428 Review
  - Algorithms & Efficiency
  - Searching & Sorting
  - Pointers
  - Dynamic Memory

- All lecture slides available on the Schedule page of the course website
  - Animated versions are on TRACS under Resources
Textbook Content

• Use the textbook to help you understand the slides
  • There will be no questions over material or code that is in the book but NOT in the slides!

• Covered sections:
  • Chapter 6.1-6.5, 6.7-6.10, 6.13 (Functions)
  • Chapter 7.1-7.4, 7.6, 7.8 (Arrays)
  • Chapter 8.1 and 8.3 (Search & Sort)
  • Chapter 9 (Pointers & Dynamic Memory)
  • Chapter 11.2-11.5, 11.7-11.8 (Structs)

• Stuff that is ONLY in lecture slides (not in textbook): Linux, Algorithms & Efficiency
Linux

- What is Linux
- Linux file system
- Basic shell commands
  - `pwd`
  - `ls`
  - `cd`
  - `mkdir`
  - `rm`
  - `rmdir`
  - `cp`
  - `mv`
  - And shortcuts, e.g. ".."
- Basic file editing (`vim`, `nano`, etc.)
- Compiling (`g++`) and running C++ programs
CS 1428 Review

• Know how to use functions to modularize code and promote code reuse
• Know how to iterate over and operate on arrays (e.g., Project 1.5)
• Know how to define and use structures
• Know how to pass parameters by reference and value
• Understand partially-filled arrays (e.g., Project 2)
• Understand arrays of structures (e.g., Project 1.0)
Algorithms & Efficiency

• What is an algorithm
• How do we analyze an algorithm's runtime efficiency

• Big-O notation
  • Go from $f(N)$ to $O(N)$
  • Given function code, state the function's runtime efficiency in Big-O
Searching & Sorting

• Searching
  • Linear Search
    • Write code, or explain algorithm, or perform algorithm
  • Binary Search
    • I won’t ask you to write the code from scratch, but I may ask you to fill in blanks in partial code, and you must understand the algorithm
  • Know best case, worst case runtime efficiency

• Sorting
  • Bubble Sort
  • Selection Sort
  • I will not ask you to write sort code, but you should be able to explain and perform the algorithms
  • Know best case, worst case runtime efficiency
Pointers

- Address operator (&)
- Pointer variables: what are they, how to define, how to initialize
- Dereferencing operator (*)
  - What it means to dereference something
- Pointers and arrays
  - Array name is address of array's first element
  - `array[index] = *(array + index)`
- Arithmetic and comparison operators on pointers
- Pointers as function parameters
  - How to pass by reference using pointer parameter
Dynamic Memory

- `new` operator
- `new` with arrays
- `delete`
- `delete` with arrays

- How to (safely) return pointers from functions
How to Prepare

• Review the slides
  • Understand all the concepts
  • Quiz yourself

• Read the textbook only to increase understanding of slides
  • Textbook has exercises/problems to solve as well

• **Review the coding assignments**
  • Fix your code if you never got it working!
  • Convince yourself your Project 2 code works

• Re-do the written homework problems, check your work against solutions on TRACS

• Review the quizzes (come to office hours if you want a quiz back)

• **Sleep! Eat breakfast!**