Final Exam Review

CS 2308
Spring 2019
Jill Seaman

Final Exam

- Thurs 5/9 11am-1:30pm
- Here (Ingram 3104)
- Closed book, closed notes, clean desk
- Comprehensive (covers entire course)
- 25% of your final grade (both classes)
- Bring your ID card!
- Bring a number 2 pencil and eraser (I will bring scantrons)
- No calculators, cell phones, headphones/earphones.

Exam Format

- 100 points total:
  - 48 pts:
    - Multiple choice (scantron form)
    - 24 questions
  - 52 pts:
    - Writing programs/functions/classes/code
    - Finding errors in code
    - 6 questions (1 per unit)

Content

7 topics:
- Unit 1 Functions, Arrays, & Structs
- Unit 2 Searching, sorting & analysis
- Unit 3 Pointers & dynamic memory allocation
- Unit 4 Intro to Classes
- Unit 5 Linked Lists
- Unit 6 Stacks & Queues
- Bonus: Linux, List ADT, Copy Constructor, Recursion
Unit 1: Functions, Arrays & Structs

- Know how to program with functions, arrays and structures.
- Passing parameters by reference (and value)
- Scope rules
- Be able to process arrays (& arrays of struct)
  - Be able to find the minimum/maximum value!
  - See review exercises
- Be able to trace code
- Be able to find errors in code

Unit 2: Searching, Sorting & Analysis

- Searching
  - Linear Search
  - Binary Search
- Sorting
  - Bubble Sort
  - Selection Sort
- See review exercises:
  - Sample exercises to demonstrate algorithms
  - Be able to code linear search and one sort

Analysis of Algorithms: efficiency

- Efficiency
  - Growth rate functions, which are faster/slower
  - Use big-O notation
  - Efficiency of
    - searching/searching algorithms
    - array access and traversal (new!)
    - linked list operations (new!)
  - See the Final Exam Review Exercises for good coverage on this

Unit 3: Pointers & Dynamic Memory Alloc

- Pointer variables: how to define + initialize
- Address of (&) and Dereferencing (*) operators
- Pointers and arrays
  - an array variable is the address of its first element
  - array[index] = *(array + index)
- Dynamic memory allocation
  - new + delete operators
  - allocate new arrays (duplicateArray, etc.)
- Pointers as parameters (call by reference, arrays)
Unit 4: Intro to Classes

- Procedural vs object oriented programming
- Encapsulation, Data hiding, Interface
- Fundamentals of classes and objects:
  - Members: variables and functions
  - private vs public
  - declaration and implementation of classes
    ‣ class declaration
    ‣ defining member functions
  - instances and the dot operator
  - inline member function definitions
  - constructors and destructors
  - arrays of objects

Unit 5: Linked Lists

- Pointers to Struct: declaration, access (s->x)
- LL Organization: nodes, head pointer, empty list, NULL
- Linked list tasks: T1-T11:
  - create empty list, create a new node
  - add to front of list
  - append to end of non-empty list
  - traversing a linked list (display, count, sum, etc)
  - how to advance 2 pointers together (n and p)
  - delete given n and p, special cases
  - insert given n and p, special cases
  - linked list destruction
- Arrays vs Linked Lists (see ListADT, last 2 slides)

Unit 6: Stacks and Queues

- ADT, LIFO and FIFO
- 4 basic operations of each data type:
  - pop
  - push
  - enqueue
  - dequeue
  - isEmpty
  - isFull
- Be able to show contents of stack or queue after a series of operations
- Be able to implement the operations (code in C++) using a static array or a linked list.
- Be able to use a driver to access a stack or queue.

C++ Programming on Linux

- Basic shell commands, know how to use
eedit, compile, run (nano, g++, a.out)
- Compiling multiple files:
  - How to split up code, what goes where
  - g++ a.cpp b.cpp
  - separate compilation
    ‣ g++ -c a.cpp
    ‣ g++ -c b.cpp
    ‣ g++ a.o b.o
  - makefile: understand the ones used for the assignments, know how to use them
Bonus
(Multiple choice questions only)

- Copy Constructor
  - Recognize copy constructor prototypes
  - Recognize declarations that call the copy constr
  - When do you need to define it yourself?

- Recursion
  - Base case vs. Recursive Case
  - What is the output?
  - Watch out for infinite recursion.

Sample Problems

See the lecture notes titled:
Final Exam Exercises
on the class website (soon)

How to Study

- Start with the topics from this set of slides.
- Use the regular semester lectures to make sure you understand the topics (quiz yourself, use the Squarecap questions).
- Use the textbook to make sure you understand the lectures about the topics.
- Do the review exercises on the Final Exam Exercises slides. Do book exercises. Practice!!
- Go over the exams and assignment solutions and quizzes (fix yours).
- Discuss with others! (and get some sleep)

Office Hours
finals week

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>5/6</td>
<td>2:00-3:00pm</td>
</tr>
<tr>
<td>W</td>
<td>5/8</td>
<td>2:00-3:00pm</td>
</tr>
<tr>
<td>Th</td>
<td>5/9</td>
<td>2:00-3:00pm</td>
</tr>
<tr>
<td>M</td>
<td>5/13</td>
<td>2:00-3:00pm</td>
</tr>
<tr>
<td>Th</td>
<td>5/14</td>
<td>10:00-11:00am</td>
</tr>
</tbody>
</table>

and by appointment