CS 200 Fall 2014
Midterm 1 Preparation Guide

Exam Date: **October 9, 2014**
Location & Time: **In-class, 9:30 ~ 10:45 AM (and in lab that week)**

In this exam, you will have a mix of multiple-choice questions and some short answer questions. To study, you should attempt the examples and exercises at the end of indicated textbook sections/ chapters. Also review the written assignments and quizzes. Your lab section during the prior week will include some practice questions from an earlier midterm.

The lab section during the week of the midterm will include a programming question. Note: the question could require you to write code, read code and/or debug code. The programming topics were stacks, queues, sorting and compound data structures (as you experienced in the programming assignments).

Key concepts

The problems in this exam will be about the concepts covered in the lectures (week 1 through week 7). Please review your lecture and lab/recitation notes (https://www.cs.colostate.edu/~cs200/Fall14/home_progress.php) The distribution of points in the midterm will reflect the amount of time spent on the topic either in lecture, in the readings, in the labs or in the assignments.

Grammars
Lecture Notes: L2
Text book: Prichard Ch. 6.2; Rosen Ch. 13.1

1. How is a grammar defined?
2. What is the difference between Types of languages?
3. How can a language be represented by a regular expression?
4. What does a derivation and derivation tree look like for a given sentence in some language?
5. What are prefix, postfix and infix expressions?

Recursion and Induction
Lecture Notes: L4
Text book: Prichard Ch. 6.1 & 6.3

1. What is the relationship between recursion and induction?
2. What is the basis step? What is the inductive step?
3. What is backtracking?
4. How can the correctness of your software be shown with mathematical induction?
5. How can the cost of an algorithm be proved with mathematical induction?

Stacks and Queues
Lecture Notes: L3 and L5
Text book: Chapter 7, 8 from Prichard

1. What is a stack? What is a queue?
2. What are the operations that are typically associated with a stack/queue?
3. How can you implement a stack/queue using arrays/references/linked lists?

Computational Complexity
Lecture Notes: L6
Textbook: Rosen Ch. 3.2-3.3, Chapter 10.1 from Prichard

1. Factors of complexity analysis
2. Growth of functions: definitions of Big-O, Big-theta, and Big-omega notations
3. What are witnesses C and k?
4. What are common forms of g(x)?
5. Be able to determine the Big-O from an algorithm description.

Sorting
Lecture Notes: L7
Textbook: Chapter 10 from Prichard

1. Sorting algorithms and their time complexities.
2. How does MergeSort work?
3. How does QuickSort work? What is the role of the pivot?
4. What is RadixSort?

Advanced Object Oriented Programming Concepts
Lecture Notes: L8
Textbook: Chapter 9 from Prichard

1. 2. What is data encapsulation?
2. What is inheritance?
3. What is information hiding and why is it useful?
4. What are interfaces and abstract classes and how are they different?
5. What is polymorphism?