CS 200 Midterm 1 Preparation Guide

Exam Date: **Feb, 24 2012**
Location & Time: **In-class**, **2:00 ~ 2:50 PM**

In this exam, you will have a mix of multiple-choice questions, some short answer questions and some questions that require you to read Java code and perform algebraic calculation. You should attempt the examples and exercises at the end of indicated textbook sections/ chapters. Many of the exam questions will have a similar form. Also review the written assignments and quizzes.

Key concepts

The problems in this exam will be about the concepts covered in the lectures (week 1 through week 5). Please review your lecture notes (http://www.cs.colostate.edu/~cs200/Spring12/Schedule.html)
The following textbook exercises and examples can help you prepare for the exam.

Recursion, Mathematical Induction, and Grammar
Lecture Notes: part. 1
Textbook : Chapter 6 from Prichard, Section 4-1 from Rosen (5-1 in 7th Edition)
Exercise: After chapter 6 in Prichard and after section 4-1(5-1 in 7th ed.) in Rosen can help you understand mathematical induction. You can start with exercises 3, 4, 5, 7, 11, 15 after section 4-1(5-1 in 7th ed.) in Rosen.

1. What is the Backtracking?
2. How are languages defined recursively?
3. Mathematical Induction
4. How is the correctness of your software shown with mathematical induction?
5. How is the cost of an algorithm proved with mathematical induction?

Advanced Object Oriented Programming Concepts
Lecture Notes: part 2.

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

Stacks and Queues
Lecture Notes: part 3 and part 4

Text book: Chapter 7, 8 from Prichard

1. What is a stack? What is a queue? How do they differ from a list?
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: part 5 -(1), (2), (3)
Textbook: Sections 3-2, 3-3, 7-1, 7-3 from Rosen in 6th Edition (section 7-1 and 7-3 are sections 8-1 and 8-3 in 7th Edition), Chapter 10 from Prichard

Examples and exercises will help you to understand concepts. You can start with exercises 1,2,7,9,10,11,12,13 from 7-3(8-3 in 7th ed.) in Rosen.

1. Factors of complexity analysis
2. Growth of functions: definitions and understanding of Big-O, Big-theta, and Big-omega notations
3. Sorting algorithms and their time complexities.
4. Know how to obtain a closed form from a recurrence relation using substitution and find its Big O complexity
5. Know how to obtain a recursive definition from the source code and find its Big O complexity
6. Know the purpose and form of the Master Theorem and how to use it to obtain Big-O complexity.
7. Know how to apply the Master Theorem to a divide-and-conquer style algorithm to find the big O complexity