

CSCI 1321 (Principles of Algorithm Design II), Fall 2003

Review for Final Exam

1 Format of the exam

The exam will be at the scheduled time for this course's final, December 15 at 8:30 am. It will consist of two parts, student presentations of their games (worth 50 points) and a written exam (worth 150 points), as described in the following two sections.

2 Student presentations

Each student will have 10 minutes to present his or her game. Each presentation should include the following.

- A discussion of the game's design, including discussion of the classes involved and how they fit into the overall design. A UML diagram would be very helpful.
- A brief description of what you learned from writing the game and what you might do differently if you were starting again from scratch.
- A demo of the game.

You should be prepared to answer questions from me and from your classmates (so allow a minute or two for that purpose).

3 Written exam

3.1 Format

You may use your textbook and any notes or papers you care to bring, but you may not use other books, each other's papers, or a calculator or computer, *except* that you may use a browser to look at Sun's documentation of the Java library.

The following are some kinds of questions that might be on the exam. It is *not necessarily an exhaustive list* of all types of questions on the exam, but should give you an idea of what to expect. The quizzes so far should also give you an idea of what kinds of questions I might ask.

- Given some Java code (possibly a complete program, possibly a fragment), answer one or more of the following questions:
 - Will it compile correctly? (You may be told that it does not.) If not, why not, and how would you fix it?
 - What does it print out? What does it (if a method) return? What value does it assign to a specified variable? Are these results correct? If not, what has gone wrong, and how would you fix it?
- Given a problem description, write a Java program, class, or method to solve it. You may be given some of the code and asked to "fill in the blanks", or you may be given descriptions of Java classes or methods to use in your solution.

- Given an algorithm, describe how its running time relates to the “problem size” — e.g., is the time required to remove a specified element from an unsorted list of N elements proportional to N , or to N^2 , or does it depend on N at all?

4 Topics to review

You are responsible for all material covered in class or in the assigned reading. The focus will be on material covered since the first exam, but there may be questions on earlier material as well, since (1) this exam is worth more points, and (2) the material is cumulative in nature. See the schedule for a list of assigned reading, but note that if you understand the summaries at the end of the chapters you are probably in good shape. You should review in particular the following topics.

- Basic Java syntax and semantics, including classes, `import`, and the difference between primitive types and objects.
- Inheritance and interfaces.
- Polymorphism.
- GUIs and graphics in Java.
- Linked-list implementation of stacks, queues, and priority queues.
- Exceptions.
- Recursion.
- Binary trees and sorted binary trees.
- Heaps, including how to implement them using arrays.
- I/O in Java — files and streams.