CSCI 1321 (Principles of Algorithm Design II), Fall 2001

Review for Exam 2

Note: The HTML version of this document may contain hyperlinks. In this version, hyperlinks are represented by showing both the link text, formatted like this, and the full URL as a footnote.

1 Format of the exam

The exam will be in class November 15. You will have 75 minutes. You may use your textbook and any notes or papers you care to bring, but you may not use other books, a calculator or computer, or each other's papers.

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.

- Given some C++ 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 function) 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 C++ program or function 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 C++ functions or classes to use in your solution.

2 Topics to review

You are responsible for all material covered in class or in the assigned reading. This exam will focus on material covered after Exam 1, but may also include earlier material. (See Homeworks and Other Assignments[1] for a list of assigned reading.) You should review in particular the following topics. Again, this list is *not necessarily exhaustive*, but should give you an idea of what topics I consider most significant.

- Defining classes:
  - "public" versus "private".
  - Ordinary member functions versus static member functions versus friend functions.
  - Overloaded functions (multiple functions with the same name).
  - Operator functions (functions that implement operators such as + or ==).

[1] `http://www.cs.trinity.edu/~bmassing/CS1321_2001fall/assignments.html`

- Dynamic memory:
  - Pointers — creation and use.
  - "new" and "delete".
- Classes and dynamic memory:
  - Copy constructor.
  - Assignment operator.
  - Destructor.
- Linked lists.