Reminders and announcements:

- The last day of class has been set aside as a review day for the Final Exam. Come prepared to ask and answer questions about the course material.
- Programming Project 1, Part B is due no later than at the start of the Final Exam, 2:45pm on Friday, December 8.

The Final Exam will be at 2:45pm on Friday, December 8. You may bring one 8.5in x 11in sheet of paper with notes on (only) one side to the exam. You may print out the sheet, but it must be in a 9-point font or larger. E.g., please do not photoreduce or print 4 pages on 1 side. If you handwrite your notes, they can be as small as you like. You may handwrite notes in the margins of a printout.

The exam will be cumulative and comprehensive. You will be expected to be able to read and write code or analyses and designs using concepts from the entire course. Emphasis will be on the material since Exam 2 in Chapters 9, 10, 11, and 14 and the material covered in lectures, exercises, and assignments through Tuesday, November 28. This material will comprise about one-half of the exam. You are not responsible for the material on files (Chapter 6), C-style strings (Chapter 8.1) and issues regarding copying, assigning, and deallocating objects containing pointer attributes (Chapter 11.4).

The format of the exam will be similar to that of previous exams, homework, and exercise questions. That is, you will be asked to write code examples, you will be asked to read and explain code examples, and you will be asked to analyze and design functions and programs to solve particular (sub)problems.

The following is a list of topics that will be emphasized, but it is in no way to be construed as an exclusive list. In addition, previous review sheets should be consulted for earlier topics.

1. Analysis, design, and implementation of main programs and functions using the following constructs.
2. Classes - design, implementation, and use. Including overloaded operators and friend functions, and vectors of class objects.
3. Recursion - what it is and how to use it; drawing execution traces.
4. Pointers - allocation and deallocation of dynamic data (both single variable and arrays); distinguishing between pointer variables, pointer values, and the "pointee"; accessing and manipulating pointer variables and pointees; drawing memory pictures.

When you are asked to write code, you will not need to write comments, function prototypes, include directives, or output formatting beyond producing newlines in appropriate places.