Reminders and announcements:

- Wednesday, April 11, has been set aside as a review for Exam 2. We will go over Homework 7 as part of the review. Thus **Homework 7 is due by the beginning of class. NO LATE SUBMISSIONS** will be allowed for this assignment. In addition, we will go over Homeworks 5 and 6, and you should come prepared to ask and answer questions about the exam material.

Exam 2 will be on Friday, April 13. 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 consist of questions on the material in Chapters 5-7 (excluding Sections 6.4-6.7, 7.4 after the Modular Implementation portion, and 7.5); material covered in lectures and in-class exercises through April 4; material in homeworks 5-7; and the Visual Basic project.

The exam will consist approximately of the following format:

- Matching section: 30 points - 15 vocabulary words matched among 20 definitions
- Questions regarding sequential and binary search, insertion sort: 20 points
- Reading algorithms in pseudocode and determining what they do: 15 points
- Writing one short algorithm in pseudocode for a simple problem: 5 points
- Multiple choice: 30 points - 10-15 questions

The following is a list of topics that will be emphasized, but it is in no way to be construed as an exclusive list.

1. Concepts of algorithms including reading and writing pseudocode. Understanding the concepts of assignment, selection (if statements), iteration (while statements), and recursion (applying a procedure to a smaller version of the problem in order to find a solution).
2. Understanding how the sequential search algorithm and the binary search algorithm work, and their differences. Understanding how the insertion sort algorithm works.
3. Programming language paradigms. Understanding how pseudocode is similar or different from a programming language. Concepts of literals, variables, constants, data types. Understanding the difference between procedures (subroutines) and functions. Understanding the difference between pass by value parameters and pass by reference parameters.
4. Concepts from Visual Basic such as TextBox, RadioButton, CheckBox, ComboBox, ListBox, MsgBox and when you might use each. You will not be asked to write any Visual Basic code, but you may be asked to read it.

5. Concepts from software engineering development such as the phases of engineering development, the waterfall model of development, incremental model of development, prototyping, open-source development, and extreme programming (XP). Understand examples of modular development and the difference between imperative structure charts vs. object-oriented collaboration. Understand forms of testing software. Purposes of software documentation. Concepts of software ownership and liability.