Test 1 will be Thursday, Feb. 17 in class. It will be closed-book and notes, except for one 8.5” x 11” sheet of paper (you can use front and back) containing any notes that you want AND the Python Summary handout. The test will cover the following topics (and maybe more).

**Chapter 11. Searching, Sorting, and Complexity Analysis**
Machine dependent measures of performance: program running time and instruction count
Machine independent measures of performance: big-oh and theta notation (definitions), orders of complexity
Complexity analysis of an algorithm to determine its big-oh notation in the best, worst, and average cases
Analysis of searches and simple sort algorithms
Recursive divide-and-conquer vs. dynamic programming to improve the complexity of an algorithm, e.g., Fibonacci and binomial coefficient
General concept of program profiling

**Chapter 12. Tools for Design, Documentation, and Testing**
Documentation at the Module, class, method, and function level in Python, pydoc
Preconditions and Postconditions, enforcement with by raising exceptions
Testing Approaches: haphazard, black-box, and white-box testing
When to test: unit, integration, acceptance, and regression testing
General concept of “proofs of program correctness”
pyunit testing in Python

**Chapter 13. Collections, Arrays, and Linked Structures**
General idea of collections and operations on collections, Abstract Data Types (ADTs) idea
Implementing Collections with arrays and tradeoffs
Implementing Collections with as linked structures and tradeoffs

**Chapter 14. Linear Collections: Stacks**
General concept of a stack: LIFO, top and bottom
Stack Operations: pop, push, peek, len, isEmpty, str
Stack Implementations: ArrayStack and LinkedStack including complexity of operations
Stack Applications: run-time stack