The Final exam will be Tuesday May 5 from 1:00 to 2:50 PM in ITT 328. It will be open notes and textbook, but no computer -- only pencil and paper. The test will focus (~60%) on topics since the last test, but about 40% of the test will be from test 1 and test 2 material. New topics since the last exam include (but are not limited to):

Chapter 10: Creating and Modifying Text
General idea of strings as an array of characters (ASCII or Unicode values) which are immutable in Python
String operations: indexing ([]), slicing([:]), concatenation (+), repetition (*), membership (in), length (len)
String methods: strip, split, find, replace, center, lower, etc.

General concept of a text file as a sequence of characters stored in a file on a harddisk
Steps for using a file: (1) `open` the file, (2) `read`/`write` from/to the file, and (3) `close` the file
File object construction with `open` and file methods: `read`, `readline`, `write`, `for`-loop to iterate each line

General concept of the directory/file structure being a tree, path, current-working directory (cwd), root, parent
`os` module functions to navigate and manipulate files/directories on disk
`os.path` module functions to check properties of files and directories on disk
`random` module function: `randint`

General concept of lists as a sequence of any type objects which are mutable in Python
List operations: indexing ([]), slicing([:]), concatenation (+), repetition (*), membership (in), length (len)
List methods: append, extend, insert, pop, sort, index, count, reverse
Parameter passing of list objects to functions via reference

The types of questions could be:
- predict the results of string or list operations/methods
- write string or list operations/methods to manipulate them is a specified way
- predict the results of several lines of code -- what text is printed or written to a text file
- writing several lines of code to perform a small task on strings, lists, or text file
- write a new function to perform a text processing task
- short answer question

Chapter 9: Building Bigger Programs
General idea of top-down design (hierarchical decomposition) and its application
Top-down design diagram showing information flow of parameters and returned values
Bottom-up implementation and testing.

The types of questions could be:
- given a top-down design diagram answer questions about it or make minor modifications to it
- perform a top-down design for a simple programming problem
- short answer question