CSCI111 2nd Exam Prep

General Topics

Everything up through the first exam (necessarily cumulative) but focus is on material since Exam 1.

Control Structures:

- relational and boolean/logical operators
- conditions and conditionals
- Indefinite loops
    - Syntax; similarities, differences to for loops

Strings

- structure
- representation (ASCII)
- common, useful methods and operations
- formatting

Lists

- structure
- creating, accessing, processing
- common, useful methods
- similarities, differences to strings

Files

- creating file objects
- reading and writing files
- handling numbers
- common methods

Dictionaries

- structure
- creating, accessing, processing
- common, useful methods
- similarities, differences to lists

Functions (also on first exam but expanded)

- use of None
- documentation strings, appropriate comments for functions
- differences between passing immutable vs. mutable data types as parameters
- putting your own functions in modules
- organizational benefits

(Not defining your own classes)

What I expect from you on exam:

- To know the Python/programming terminology
- To be able to read a program and describe what the program is doing at a high level in plain English, trace through the program's execution given input (control flow), and say what the program outputs
- To be able to write a program (given an algorithm or creating your own algorithm, given a problem)
    - Syntax must be very close to correct (correct keywords, indentation, special characters, variable naming, operations)
    - Since it's on paper, there is some leniency—you may mark up your exam somehow if, for example, something should be indented
    - No need for constants or comments on an exam *unless specifically requested*

Suggestions on how to prepare:

- Practice programming on paper and verify program in Python.
    - Use problems from class, labs, or textbook.
    - Some problems from class suggest implementing again in a different way or occasionally there are problems that we didn't do in class.
    - Do the practice/interactive exercises in the textbook, although they often aren't at the appropriate level of difficulty for an exam.
    - The important part is that you practice solving problems. Given a problem, how can I solve it using the variety of control structures, data structures and their operations, methods, etc., available to me?
- Practice reading through programs, tracing through them, and saying what the output should be
    - The interactive book is helpful for showing you what happens when you run a program. Make sure you state what you think should happen first, before looking at what actually happens. (So you're not just like, "oh, yeah, that's what I thought." Critically think about the program *first*. Write out what happens.)
- Read through slides for vocabulary, review questions, and non-problem-solving exercises.
- Use techniques on other problems. For example,
    - Refactor code to use functions (lots of problems where you could "functionalize it")
    - Write test cases using test.testEqual for functions

Some practice problems:

- Modify your lab code to find the x^{th} most popular name
- Modify the children's book lab problem to take a string and, for each word in the string, maps the first letter to the word (like a children's book)
    - Other modifications: read from a file; make a function that returns the dictionary

The goal is to practice. You know so much, and there are many different ways to apply your knowledge to solve a variety of problems.