Intro

Why take this course?

CS151 is great fun. 150 (or whatever your first course in programming was) is essentially an exercise in logic. 151 is much more. Now that you know how to write programs, this course will show you how to use this knowledge to do useful and non-obvious things. When students go on job interviews, the questions they are asked usually concern 151 material.

What we cover

The first two weeks of the semester will largely be devoted to transitioning you from Python to Java. For the rest of the term we will look at the Java Collections Framework (ArrayLists, Stacks, Queues, Binary Trees, AVL trees, HashMaps, TreeMaps, etc.) Each week we will take one of these structures, implement it from scratch, and then write an application program with it. By the end of the term you will know the Collections well, and will be able to implement just about anything you can think up.

Course Structure

We will have

- 2 in-class exams
- A final exam (on Thursday, May 11 at 9AM)
- 10 labs (and prelabs)

Clickers

- We will be using clickers this semester. Most of you should have them from 150 last semester.
- 5% of your grade is for "attendance and participation". Here's how I measure that. If you use your clicker in 75% of the classes with clicker questions you get all 5 participation points. If you use it in 60% of the classes you get 3 points; 50% gets you 1 point.
- The only thing we will use Blackboard for is registering your clicker.

The Labs

The labs are where you will do most of your learning. The lab sessions are on Monday afternoons (2:30 to 4:30) and Tuesday afternoons (1 to 3). The lab sessions will only get you started. You will probably spend 8 to 10 hours on each of the labs, so schedule your time well.

Due Dates

The labs are due on Sunday at 6PM. You will be turning them in online and the handin program puts a time stamp on them, so we will know when you handed in your code.

The prelabs are due in class on Monday. If you don't come to class, send them to me in email by noon on Monday.

Late Work

I won't accept late prelabs.

There are 3 stages of late work for the lab:

- Labs that come in by 9AM on Monday are docked 10%.
- Labs that come in by 6PM on Wednesday are docked 50%.
- Labs that come in after 6PM on Wednesday are docked 100%.

Extensions

Sometimes you get sick. You may use up to 3 Late Lab Extensions that allow you to hand in a lab up to Wednesday at 6 without penalty. To use one include a text file in the folder you hand in called LateLabExtension.txt. In the body of the file say which extension you are using (#1, #2 or #3)

Note that I want lab material to come in by Wednesday under any condition. If you are so sick you can't get out of bed for several days let me know. The LLEs should be enough to handle the usual colds, flus, etc. Save them for when you need them rather than for days when you aren't organized enough to get the labs done on time.

Partners

In all of the labs you are welcome to work with a partner during the 2-hour lab session. If you intend to have a career in this field you need to learn to work effectively as part of a team, so find a buddy. The first three labs you need to complete on your own. Most of the remaining labs you can complete with your partner. Any file you hand in should have a comment at the top listing all of the authors.

If you work with a partner on a complete lab you two should hand in one copy, under either of your usernames. Include a text file Partners.txt that explains who worked on the program.

The Honor Code

This should be very straightforward:

- Don't cheat on the exams
- Be honest about who worked on the labs you hand in.

The Text

Data Structures & Problem Solving using Java by Mark Weiss

You need something to read; you can't learn all of the material in this class from the lectures. You will probably find this challenging to read at the start; it should get easier as you get more experience.

Why Eclipse?

We will be using an IDE (Integrated Development Environment) called *Eclipse*. It is more complex than the tools most of you are used to using. Here is why we use it:

- Eclipse is great at handling programs built from multiple class files. Real programs use a lot of files. For example, the project from my Compilers course is made up of about 75 files. Handling a program like this without an IDE is a nightmare.
- Eclipse knows about Java and can save you time that you won't have to spend looking through documentation.
- Eclipse can help you integrate unit testing into your development cycle.