Syllabus

Introductory Programming **Allen B. Downey**
Fall 2004

- Professor: Allen Downey, allen.downey@olin.edu, Olin Center 359, x2558.
- Class time: Tues, Thurs, 8:30 to 9:20
- Textbook: Downey, *How to Think Like a Computer Scientist*.
- Web page: The class web page is at `http://whiteboard.olin.edu/ip`. On campus, it can be abbreviated `wb/ip`.
- Class mailing list: `http://lists.olin.edu/mailman/listinfo/intro_prog`

Description

You will not be surprised to hear that Introductory Programming is an introductory class that teaches basic programming. It is a 2-unit class intended for students with no programming experience (or very little). The goal is to give incoming students a chance to learn the programming concepts and skills they will need in other classes.

The language of instruction this semester is Python, which I have found to be an excellent "first language." It is also similar to Matlab, which is used in several other classes at Olin.

Coursework

This class consists of a series of modules; each module includes reading from the textbook (and other sources), a lab exercise intended to allow you to practice the ideas in the reading, and some kind of evaluation, either written, oral or practical.

We will spend some class time going over the reading and working together on practice exercises. Some class time will also be free for you to work on the lab exercises and evaluations.

At the end of each module, you will have several opportunities to pass each evaluation. The class is self-paced in the sense that you will have some ability to get ahead, or to catch up if you fall behind. The only limitation is that some in-class activities are necessarily synchronous.

Most students will take this class P/NR. The threshold for passing will depend on the number of evaluations a student attempts, including repeated attempts at the same module.

Communication

The easiest way to reach me is by email; I read my mail fairly often, even in the evening and on weekends. Unfortunately, my email accounts are constantly hammered by an astounding amount of spam, which means that I sometimes lose email, either because it gets caught in a spam filter or because I delete it without recognizing it as legitimate. So, if an email goes unanswered, please give me another chance. You are also welcome to call me at x2558 any time, or at home during normal waking hours.

If I am in my office and my door is open, you are welcome to come in and talk to me. If I can't meet with you, I will let you know, and schedule a meeting for later. When I know my schedule for the semester, I will make it available.

You are encouraged to communicate with the other members of the class using whatever means you choose, including the class mailing list. You can join the mailing list (and change your membership configuration) at `http://lists.olin.edu/mailman/listinfo/intro_prog`. It is appropriate to use this mailing list to discuss anything pertaining to the class, or to the topic of the class, broadly defined. For example, I will probably post messages about news items that pertain to computer science or technology and society.

Collaboration

You are encouraged to work with other students (in and out of this class) as you develop your understanding of the material. You are free to work together on lab exercises, but remember that these exercises are your opportunity to practice the material, so make sure you are not depending on your partner too much! Most of the evaluation exercises will be individual efforts, although we might have some that are collaborative.

As always, your actions in this class are bound by the Honor Code; in particular, the principle of integrity states, "Each member of the college community will accept responsibility for and represent accurately and completely oneself, one's work, and one's actions." When you submit course work for evaluation, you are representing that the work is entirely yours, unless you state otherwise. Representing someone else's work as your own is a very serious violation of the Honor Code.

Topics

The following are the topics we will be covering, roughly in the order we will be covering them:

1. Variables and expressions. Amoeba World.
2. Statements and functions. Turtle Graphics.
3. Conditionals, fruitful functions.
4. Encapsulation, generalization.
5. UML state diagrams.
6. Iteration, strings.
7. Lists, tuples, dictionaries, textual analysis.
8. Files, exceptions, classes, objects.
9. UML object diagrams.
10. Classes, functions and methods.
11. Collections of objects.
12. Object-oriented programming, inheritance.
13. UML class diagrams.