Syllabus

Introductory Programming **Allen B. Downey**
Fall 2006

- Professor: Allen Downey, allen.downey@olin.edu, Olin Center 359, x2558.
- Class time: Tues, Fri 9-10am.
- Textbook: Downey, *Introductory Programming in MATLAB*.
- 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 MATLAB. The exercises we will work on are designed to supplement the ICBs and prepare students for the programming work in those classes.

Coursework

This class consists of a series of 10 modules; each module includes reading from the book (and other sources), a lab exercise intended to allow you to practice the ideas in the reading, and some kind of evaluation, either written 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. To pass the class, you must pass 8 of the 10 modules and make a good faith attempt at all 10. Occasionally, a student may complete a module but not demonstrate command of the material. In that case I may ask for additional evidence, usually by oral or practical evaluation.

Communication

The easiest way to reach me is by email; I read my mail fairly often, even in the evening and on weekends. 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. Programming and debugging.
2. Formal and natural languages.
3. Variables, values and workspaces.
4. Functions: calling, defining, parameters, arguments, return values.
5. Floating point, relative and absolute error.
6. Root finding, Newton's method, fzero.
7. Iteration: for loops and while loops.
8. Conditional statements.
9. Matrices: creating, accessing, modifying.
10. Time-based discrete simulation.

11. Euler's method, ode45.
12. Second order systems. Multidimensional systems.