Syllabus

Software Design
Fall 2004

Allen B. Downey

- Professor: Allen Downey, allen.downey@olin.edu, Olin Center 359, x2558.
- Class time: Mon, Tues, Thurs, 12:00 to 12:50.
- Lab times: Friday 8:00 to 10:00 am OR 11:00 to 1:00 pm.
- Textbooks: Downey, *How to Think Like a Computer Scientist*, plus your choice of another Python book.
- Web page: The class web page is at `http://whiteboard.olin.edu/sd`. On campus, it can be abbreviated `wb/sd`.
- Class mailing list: `http://lists.olin.edu/mailman/listinfo/software_design`

Description

Software Design is an introductory class that teaches basic programming. It is called "Software Design" because it is based on the idea that programming is more than just translating a well-understood solution into code; it is the process of solving a problem by designing a language for expressing the solution.

This semester, we will be using the Python programming language. In other semesters, this class is taught using Java. I believe that Python is an excellent first language for beginning programmers, but it also offers advanced features that will be exciting for people who know other languages.

During the first 8 weeks, you will work on series of exercises covering topics in graphics, animation and graphical user interfaces; network and concurrent programming, and object-oriented design. During the next 6 weeks, you will work in small groups on an extended project that applies software design to an application or problem of your choice. In previous semesters, projects have included network applications and games, and tools for creating and manipulating images and sound.

Regular class meetings and lab times will be used for lectures, hands-on work and collaborative work, as well as written exercises and other forms of evaluation.

Students with no programming experience and students with background comparable to the CS AP should both find this course interesting and worthwhile.

Coursework

Work in this class will include readings from the textbook and other sources, weekly programming homeworks, a final project, two one-hour exams, a final exam, and written quizzes.

The total course load is intended to be 12 hours per week (including class time); the load should be spread evenly across the semester.

Final grades are determined by a weighted average of exam scores, quizzes, homeworks, and an additional factor that reflects my subjective impression of the quality of your work, your progress and effort, and your contribution to the educational goals of the class.

- Quizzes: We will have about one quiz per week. The quizzes are intended to both me and the students figure out how things are going. They make up a small part of the final grade. Since there are many quizzes during the year, it is too complicated to try to schedule alternative dates. Instead, I will drop the lowest quiz score at the end of the semester.

- Exams: During the first week or so, we will choose dates for the two midterm exams. Both exams will be given during a regular class meeting and will take 50 minutes. The final exam will be given during the final exam period.

  If you miss a midterm for an extremely legitimate reason, your final grade will be based on the other exams. If you miss a midterm without a legitimate reason, you will receive a zero. If you miss two midterms or the final for any reason, you cannot pass the class.

- Homeworks: During the first 8 weeks of the class, we will have weekly homeworks that are meant to give you an opportunity to apply and practice the material we cover in class. Homeworks are graded on a coarse scale: check-plus has the numerical value 10/10; check has the value 8/10; check-minus has the value 6/10.

  I believe that these homeworks are most effective if we have an opportunity to discuss them in class shortly after the due date. In order to make that possible, I will have to enforce the due dates with some strictness. Homework that is up to one day late can get a check or lower; homework that is more than one day late can get a check-minus or lower. But late is still much better than never!

- Project: During the last 6 weeks of the class, you will be working on a project. I will provide more information about the projects later.

- Competency assessment: I hope that your activities in this class will help you to develop many of the competencies that Olin has identified as most important to your college education. The two competencies that are most central to this class, and which I will assess, are design and diagnosis. In addition, we will have one assignment intended to encourage you to think about life-long learning.

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/software_design`. 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

Some of the coursework you will do this semester is intended to help you develop understanding of the material, and some is intended to allow me to evaluate your understanding. While you are developing your understanding, you are encouraged to work with other students, but when you are being evaluated you will have to work alone (with the exception of the project, which is 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, expressions, statements and functions. Turtle graphics.
2. Conditionals, fruitful functions, encapsulation, generalization. UML state diagrams.
3. Iteration, strings, lists; text processing.
4. Tuples, dictionaries, textual analysis.
5. Files, exceptions, classes, objects. UML object diagrams.
6. Classes, functions and methods; data structures.
7. Collections of objects, probability and statistics.
8. Object-oriented programming, inheritance. UML class diagrams.
9. Graphical user interfaces.
10. Threads, concurrent programming.
11. Remote objects, distributed programming.