CS 40 Handout 0: Syllabus September 1, 2015

Instructor: Art Lee (http://www.cmc.edu/pages/faculty/alee/)
Office: 216 Adams, Phone: 7-0410, Email: alee@cmc.edu
Office hours: Tue 4pm-5pm, Thu 1pm-2pm, and by appointment.
Tutoring hours: TBA on the course web.
Lectures: Tue, Thu 9:35am-10:50am, Roberts North 15

Course Overview

The focus of the course will be on teaching you how to begin with a problem statement and then systematically design a computer program that solves practical computational problems by programming the Web and writing data analysis programs in a modern high-level language, Python. You will learn languages, tools, and techniques for developing interactive, dynamic web content. For data analysis, assignments solve real data manipulation tasks from science, business, and the humanities, enabling you to solve such problems in your own discipline. This course is intended for students who do not plan to major in computer science.

Throughout the course, you will learn many fundamental concepts vital in the study of computer science through learning the programming language Python, web programming, and data programming. You will learn languages useful for web development such as HTML, CSS, and JavaScript. More specifically,

- You will learn Python programming, HTML, CSS, JavaScript during the first month and a half or so.
- You will then start more serious web programming to create dynamic web contents with both client- and server-side scripting using a Python web programming framework such as Flask with a database system such as SQLite. You will build a rather sophisticated website of your own.
- In the latter part of the semester you will write Python programs using some powerful libraries such as NumPy and matplotlib to analyze and visualize data that you find on the Internet, e.g., stock market data, presidential election results, Shakespeare’s Hamlet, etc.

There is no prerequisite for this course. I assume no prior programming or computer science experience. We will start from the very beginning and move at a rather fast pace.

The primary source of information about the programming language Python and the fundamentals of computer science that we will study this semester will be my notes and Think Python: How to Think Like a Computer Scientist, Version 2.0.16, by Allen B. Downey, 2012. For Web programming the main source of information will be references on the Internet on HTML, CSS, JavaScript, and jQuery and from Flask: Web Development, One Drop at a Time and some other sources from the Internet, which I will tell you about when the time comes. For Data Programming, I will extract some material from the Internet. I will tell you a little more about these references and you can decide to buy any of them or not afterwards.
You will be using a good subset of the programming language Python using a programming environment called Canopy. You will find instructions on the course website on how to install it on your computer.

**Pragmatics**

We will meet for lecture on Tuesdays and Thursdays for 75 minutes each. In lecture, I will make use of both the whiteboard and of a projected laptop computer. I will use the laptop to view/create programming examples, and after lecture I will post those examples on the course website.

In each lecture you will be given a reading assignment for the following lecture. Almost every week I will post a programming assignment for the following week on the “Course Schedule” page of the course website. It will be due about a week later by electronic handin. For the exact details about the programming assignments see the schedule page on the course web. Use the “Sakai” link on the course web to hand in your assignments.

There will be a link on the course web page to the lecture schedule. This schedule will list the topics and reading assignments for each lecture. Following each lecture, I will update this schedule to reflect what was actually covered in the lecture that day. By the end of the semester, the schedule page will contain a record of almost all of what we have covered. Note that what is on the page will not replace all the things that we went over in class though. So, it is imperative that you attend every lecture to see everything that we discuss.

There will be two exams and a final project. See the schedule page for the exact times and places of the exams. Makeup examinations will only be given for verified, officially sanctioned college activities. All makeup examinations may be oral.

Your final grade will be calculated from exams, problem sets, final project, and class participation. For exact details see the “Grading” link on the course website.

**Programming Assignments**

Most of the programming assignments will use Python although some will use other languages such as HTML, CSS, JavaScript, and jQuery. For Python programming we will use a programming environment called Canopy. Both Python and Canopy are free for you to install on your computer. Installation links and instructions will be available on the course web. If you don’t have a computer of your own to use, please let me know as soon as possible.

**Topics To Be Covered**

See the ”Course Schedule” page on the course web.

**How To Approach This Class**

For most of you, this will be your first class in computer science. Here is some advice on how to approach this class.

- Skim the relevant chapter(s) in the suggested reading before you come to lecture, and read more carefully after the lecture. The lecture will not assume that you have read the suggested chapter(s) in advance, and the lecture should help you understand the context and key points of the chapter. Nevertheless, the lecture does not fully replace reading, because the lecture may omit fine points that are discussed in the reading assignment.

- Concentrate in lecture. The concepts that are presented in lecture are what’s important; you will be able to find the details in the book or in my lecture notes. Writing down everything that is said and then trying to figure it out later would not be a good way to learn. Instead, think about what is being said. I will ask many questions. Try to answer all of them even if you do only in your head. Raise your hand and ask a question when you don’t understand something. Try to understand everything. Don’t give up!

You are welcome to ask as many questions as you wish, but try not to answer too many of my questions.
This is to give others a chance to answer them too.

- Respect the assignments. Some students expect that the assignments will be straightforward if they have done the reading and concentrated in lecture. Not so! The assignments are designed to challenge you by requiring that you apply the concepts you have learned to new situations. The assignments will be your most important learning experience in the course. They will rarely be straightforward. First couple assignments may seem straightforward but that is not a good indication of what the remaining assignments will be like. Assignments become more complex as we incorporate more concepts and ideas into them throughout the semester. Here is a suggestion on how you might approach each assignment:
  
  - As soon as you receive an assignment, read it and understand what the problems are and what is being asked, even if you are not ready to commit any time to actually solving them. Ask questions if the specification is not clear. Once you have the problem in your head, it is most likely that you will find yourself thinking about it, perhaps at least subconsciously if not actively.
  
  - Ask questions when you get stuck. I am available via email and during the office hours. Your tutors are available during their office hours as well.
  
  - Start early so that you will have time to take a break when you get stuck. Waiting until a few days before the due date is a bad plan. When I give you a week, I give you that much time because it would take that much, factoring in whatever else you would be doing in a typical week. It gets harder to get help from us as it gets close to the due date – plus you won’t have time to take a break by then.
  
  - Study the solutions. Your struggles on the assignment will be for naught if you don’t review your work. Getting the right answer isn’t enough. There is almost always a better way to solve a problem in programming. See if the sample solutions that I give you are better than yours; if so or if not, convince yourself why and how?

Students are often surprised by the amount of human effort that has to go into designing, writing, and testing a program. Complaints from students about the amount of time required by introductory computer programming courses are universal. You should expect to spend three hours outside of class for every hour that you spend in class. In other words, you should expect to spend nine hours per week reading, studying, and developing programs. Some of you will spend more than that. Please keep this in mind when setting up your schedule for the semester!

**Getting Help and Information**

You can get to the course web page from Sakai. It will contain a variety of information resources, including course schedule, our consulting hours and email addresses, tutors’ consulting hours, links to course handouts and problem sets, links to examples from lecture, and links pertaining to the textbook, Python, Canopy, HTML, CSS, JavaScript, jQuery, Flask, etc.

I will set up a mailing list so that I will be able to reach everyone in class.

The Announcements link on the course web page will be used to make general announcements for everyone registered for the course. I will try to reach everyone by using the Announcements link until the mailing list is set up. After that I will communicate with you via email.

Announcements specific to a problem set will be made in the FAQs link associated with the specific problem set. If you have a question on a specific problem set, please consult the FAQs link associated with it before you fire up an email with a question. It is likely that your question has already been answered in the FAQs.

Please check the announcements, your email, and the FAQs often. If any corrections or changes are to be made on an assignment, it will be in the FAQs link for the assignment.

I encourage you to see me when you need help, advice, or encouragement. I will always be available during my regular office hours each week, and you may also make appointments for other times. Simple questions can often be answered by phone or email.
Cooperation vs. Cheating

Working with others on assignments is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we will permit.

When working on programming assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a programming problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a high-level discussion of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, you are cheating.

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

*We do not distinguish between cheaters who copy others' work and cheaters who allow their work to be copied.*

If you cheat, you will be given an F in the course and referred to the appropriate college office. If you have any questions about what constitutes cheating, please ask.

Students With Disabilities

Reasonable accommodation will gladly be provided to the known disabilities of students in the class. Please let the instructor know of such situations as soon as possible.