Term: Winter 2008
Lectures: Monday, 3:00 pm – 6:00 pm
Location: 4151 USB
Labs: Wednesday, 5:00 pm – 7:00 pm
Location: 1250 USB
Instructor: Dr. Charles Severance
Email: csev@umich.edu
Telephone: see “Email”
Office hours: by appointment

Course Description

This course will introduce basic programming skills with a focus on data analysis and visualization.

Course Web Site

The course will be administered via CTools at https://ctools.umich.edu/portal. When you log in to CTools you should see two tabs for the course. One tab is titled “SI 182.” This is where much of the course interaction will take place. The second tab is titled “SI 182 W08.” This is where assignments will be distributed and graded. You should see both tabs; if you do not see both tabs (after looking in the "more sites" drop-down) be sure to contact the instructor to ensure that you are properly enrolled in the course.

Course Mailing List

Much of the communication for the course will accomplished via email. Students are expected to read the email that comes to them from the course. The mailing list is also a place to get help in the class. It is completely acceptable for a student in the course to attempt to help another student over the mailing list. The Instructor will read all mail and correct any incorrect advice that one student gives another. The email address for the course is si182@ctools.umich.edu. All students have permission to post to the mailing list.

Required Books

The following textbook is required for the course:

Python Programming: An Introduction to Computer Science by John M. Zelle
Published: Franklin, Beedle & Associates, Inc., 2003
ISBN: 1887902996

Required Tools

Students in the course will use a number of tools including Python and a text editor. It is possible to complete all the work in the course using 100% free tools. There are sufficient free tools to do the work on a PC or a Macintosh.
Optional Tools

It is highly recommended that every participant in the course own and bring a laptop to each class session. This is not a requirement, and all of the work in the course can be completed on a desktop computer or on lab computers. However, you will find parts of the course more valuable if you can play with code examples as we demonstrate them in class. A laptop will also make it easier for you to work on your assignments and projects in your spare time.

For backup purposes I suggest you buy a flash drive with at least 2GB of storage space. This will also allow you to store applications you use in case you are without a laptop and available computers do have what you need installed.

While the instructor likes his Macintosh laptop very much, you can do the course on either a PC or a Macintosh. Where there are significant differences when completing an assignment on the two systems, examples and documentation will be provided for both systems.

Giving and Receiving Assistance

The first time you learn technical material it is often challenging. We are going to cover a wide range of topics in the course and we will move quickly between topics. Because it is my goal for you to succeed in the course, I encourage you to get help from anyone you like, especially in the portion of the course before the midterm and even for the completion of assignments.

However, you are responsible for learning the material, and you should make sure that all of the assistance you are getting is focused on gaining knowledge, not just on getting through the assignments. If you receive too much help and/or fail to master the material, you will crash and burn at the midterm when all of a sudden you must perform on your own.

If you receive assistance on an assignment, please indicate the nature and the amount of assistance you received. If the assignment is computer code, add a comment indicating who helped you and how. If you are a more advanced student and are willing to help other students, please feel free to do so. Just remember that your goal is to help teach the material to the student receiving the help. It is always appropriate to ask for and provide help on an assignment via the course-wide mailing list.

Classroom Rules

We are all here to learn. I like a relaxed classroom where everyone feels comfortable. You are welcome to bring drinks or snacks to class (assuming it is allowed in the room we are in). You can (and should) bring your laptop to class. You can check email or MySpace or whatever as long as your keyboard clicks are not distracting. I would rather have you come to class and listen with one ear than not come to class at all. As a courtesy to others, be sure to put your PDA and cell phone on silence/vibrate. Coming late to and/or leaving early from class is fine as long as you don't disturb your classmates. I sometimes forget to schedule a break during a long lecture, so feel free to suggest a break if it appears that I have forgotten to do so. Ask questions at any time, and if you have some expertise in a particular topic, feel free to raise you hand and share it with the class. I will not be offended and in fact am here to learn just like you are. Sleeping in class is OK too, but I will do my best to keep you awake for the whole class period. Our primary purpose in the classroom is to interact and learn from each other.

Assignments
There will be weekly assignments throughout the course. Regular assignments allow you to learn the material in small "chunks" and to keep a close eye on how well you are understanding the material.

Every assignment will have a due date that will be strictly enforced. The assignments are often used as examples in class, and we need to be able to openly discuss an assignment as quickly as possible. For this reason late assignments will receive a zero. CTools will accept assignments after the due date, and you can turn in a late assignment or revise a previous assignment in order to receive feedback, but no points will be given for an assignment that is turned in after the due date.

Exams

There will be two midterm and two final exams. The first of each is a traditional written exam intended to measure mastery of the course material, including programming knowledge; the second is a practical exam, administered during a scheduled lab session, intended to measure programming skill.

Grading

The graded work in the course will be weighted roughly as follows to determine a final percentage grade:

- Homework: 50%
- Exams: 40%
- Class participation & instructor discretion: 10%

Grades will be awarded as follows:

- A+ 97% ← You have to work really hard to get this
- A  92%
- A- 87%
- B+ 82%
- B  77%
- B- 72%
- C+ 67%
- C  62%
- C- 57%
- D  52% ← You also have to work really hard to get this
- F  47% ← Or this

Course Outline

The course effectively consists of two parts. In the first part we march through the textbook in quick fashion. We might even skip bits and pieces here and there so we can move quickly. In the second part, once we have learned the basics of programming, we will focus on data analysis and visualization.

Since this is a brand new course and the first time I am teaching Python, we will keep some flexibility in the following schedule. If it seems like we need to spend more time on a particular topic, we will shuffle the schedule. It is important for you to let me know when you are having problems. This will help me pace the material appropriately and cover some material in greater depth.
I am perfectly happy to loop back and review material if it appears we went through something too quickly. Just let me know when you need me to do this.

<table>
<thead>
<tr>
<th>WEEK</th>
<th>DATE</th>
<th>TOPIC</th>
<th>REQUIRED READING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>January 7</td>
<td>Introduction to the Course and Computers</td>
<td>Chapter 1</td>
</tr>
<tr>
<td>2</td>
<td>January 14</td>
<td>Writing Simple Programs / Numbers</td>
<td>Chapters 2 &amp; 3</td>
</tr>
<tr>
<td>3</td>
<td>January 21</td>
<td>MLK DAY – NO CLASS</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>4</td>
<td>January 28</td>
<td>Computing with Strings</td>
<td>Chapter 4</td>
</tr>
<tr>
<td>5</td>
<td>February 4</td>
<td>Defining Functions / Decision Structures</td>
<td>Chapters 6 &amp; 7</td>
</tr>
<tr>
<td>6</td>
<td>February 11</td>
<td>Loop Structures</td>
<td>Chapter 8</td>
</tr>
<tr>
<td>7</td>
<td>February 18</td>
<td>Midterm Exam</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>February 25</td>
<td>WINTER BREAK - NO CLASS</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>March 3</td>
<td>Loop Structures (continued)</td>
<td>Chapter 8</td>
</tr>
<tr>
<td>10</td>
<td>March 10</td>
<td>Reading and Writing Files</td>
<td>TBD</td>
</tr>
<tr>
<td>11</td>
<td>March 17</td>
<td>Parsing Data</td>
<td>TBD</td>
</tr>
<tr>
<td>12</td>
<td>March 24</td>
<td>Visualization</td>
<td>TBD</td>
</tr>
<tr>
<td>13</td>
<td>March 31</td>
<td>Visualization</td>
<td>TBD</td>
</tr>
<tr>
<td>14</td>
<td>April 7</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>April 14</td>
<td>Last Day of Class</td>
<td></td>
</tr>
</tbody>
</table>

**Success in the Course**

This is a fast-paced course and it covers a lot of interesting topics. The course is designed for students with no programming experience. If you stick with the course and invest the necessary time, you will be amazed at how much you will learn in 15 weeks.

If you do not have any programming experience, some concepts will take some time to sink in. Do not worry too much if you feel like you are in a fog at times. The assignments are the best way to track your progress through the material.

Usually the biggest problem students encounter in the course is trying to do everything in a few hours right before an assignment is due or right before an exam. If you only think about the course a few hours each week, you will get some of the details but they will not mesh together to provide the big picture. Programming is easy once you get the big picture. The textbook will become an easily scanned reference for you once you know what to look for and why you are looking for it.

Cramming does not work very when dealing with the material in this course. This is because the material in the course is actually very easy once you "get it" – once you understand some basic principles. No amount of memorization will make up for not having the big picture. Try not to get stuck on any one thing – it is all easy once you "get it." If you do get stuck on something and feel like you are going in circles, ask for help, look at something else, or come at the problem from a different direction.

Good luck and welcome aboard!