About CS100

Revision Date: April 15, 2015

1 Introduction

CS100 takes a pragmatic approach in teaching programming. Basically, the only way you truly learn a subject is by practice - lots and lots of practice. This is not a course where you can sit passively and listen to a lecture and then be able to do what we expect from students in the course. You have to take a proactive approach to the material in this class, and learn how to dig out information and details on your own.

If you pursue a degree in computer science, you will be writing code for many years to come. We can’t tell you what the programming language of 2030 will be, but we can help ensure that you have the ability to learn new materials on your own. This class is an important experience for life-long learning. The skills and abilities that you gain in CS 100 should be of use to you for many years to come.

2 Rewards for hard work

As mentioned in the previous section, you can’t get better at programming by watching your instructor or fellow students write code. You only improve when you take the time to program yourself.

This class will put you in unfamiliar situations (potentially new operating systems and new programming languages). We expect you to take the time and energy to get comfortable with these systems. The greater the variety of systems and languages to which you gain exposure during your education, the better your overall perspective.

Basically, you will get out of this class what you put into the class.

3 Why Linux

The big three operating systems run on most computers are Microsoft Windows, MacOS, and Linux. Linux and MacOS are versions of the UNIX operating system; MacOS is tightly controlled and rather expensive, while Linux is free and open source. Approximately 90% of all smart phones run either IOS or Android, both of which are derived from UNIX. In addition, 50% or so of all corporate servers run Linux. Thus UNIX is an important piece of software with which all Computer Scientists ought to be familiar.

UNIX and especially Linux were designed by programmers for programmers, while Microsoft Windows was designed for people with little experience using computers. This makes Windows a non-ideal system under which to build large software projects. In fact, rumor has it that the Windows OS is built using a set of high-powered UNIX machines. We figure most students are passably familiar with Windows, so CS100 uses Linux as an environment for teaching programming.
4 Why C

C is the most popular language for building software projects, followed by Java, C++, and Python. The syntax of nearly all modern programming languages are derived from C. Thus, it is relatively easy to transition from C to any of these other languages. C does have some tricky bits; this is why CS100 uses a set of coding conventions to help students navigate these issues.

5 Why Vim

An editor is a word processor for programmers and is used by programmers to write and modify source code. Source code refers to the words and punctuation that computer programs start out as. With regards to editing, a programmer’s primary concerns are entering source code quickly and modifying source code even more quickly. There are two great and timeless editors geared especially for proficient programmers: emacs and vi (nowadays vim).

In this class, you will learn vim (although you can also choose to learn emacs if you wish).