Introduction to Computer Programming

# The Course

This is course for those who have little or no experience in computer programming. The students will gain insight into programming ideas and goals to help decide if further study might be for them. The instructor will introduce some of the central ideas, techniques, and styles he has gained over his career. A primary goal will be to get the students to do actual programming. This is where the actual “feel” of programming is experienced. Students must bring their laptop computers and have rudimentary skills such as powering on, creating and modifying text files, and browsing the Internet.

# The Instructor

Ray Smith (Charles Raymond Smith) - [raysmith619@alum.mit.edu](mailto:raysmith619@alum.mit.edu)

* Years of Programming – engineering, scientific, financial
* Programming environments – embedded, systems, application
* Languages – C, C++, Perl, Java, Python, Assembly, shell, Fortran, PL1, …
* Currently investigating graphics / game programming

# Course Outline

1. Introduction
   1. Goals
      1. Understanding, empathy
   2. Instructor
      1. Ray Smith (Charles Raymond Smith)
      2. Email - [raysmith619@alum.mit.edu](mailto:raysmith619@alum.mit.edu)
      3. 50+ years of programming
   3. Requirements
      1. Interest
      2. Laptop computer
2. What is computer programming?
   1. Similar activities
      1. Recipes - cooking, Instructions
      2. Construction - Roads, buildings)
   2. Why should I care?
   3. Alternatives – e.g. get someone else to do it for me.
      1. Specifications – plan, restrictions, goals
3. Some simple building blocks of computer programs
   1. A Computer and it Parts
      1. Output
         1. Display
         2. Printer
      2. Input
         1. Keyboard
         2. Mouse
         3. Other cool stuff
      3. Arithmetic – adding, subtracting, …
      4. Memory – where we store stuff (data)
   2. Language Parts
      1. Access / Manipulation of the computer parts
      2. Move, Change, Compare, Decide
   3. Computer Programming Languages
      1. Python
      2. Java
      3. C / C++
      4. Assembly Language
4. Let’s Write a Program
   1. I’m thinking of a number…
      1. Traditional “20 questions”
      2. Program picks number and player guesses: “Is it greater than 5” continually.
      3. Main player choices are “> N”, “<N”, “=N” where N is a number, and “Q” for quit
   2. Goals – What’s it going to do? Or at least, what would be nice?
      1. Elaborate the goals of our program
      2. Prioritize – e.g. No 3D graphics display first round 😊
   3. The Plan – How are we going to make it do what we want?
      1. Or at least, some of what we want?
      2. Displaying Results/Request