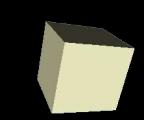
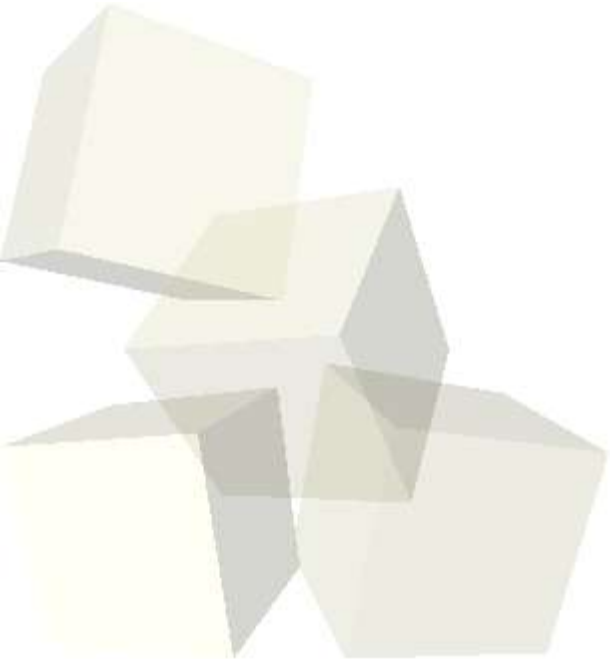
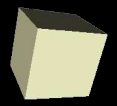


Logic and Conditional Execution

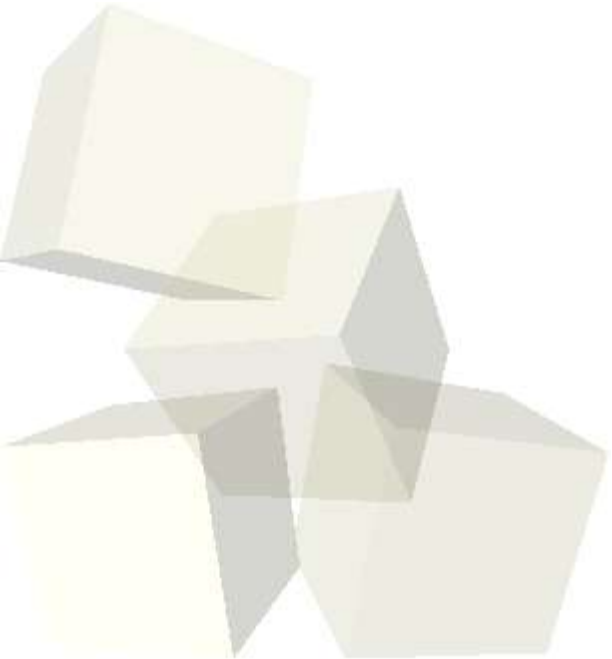
8-29-2005

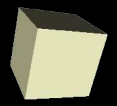




Opening Discussion

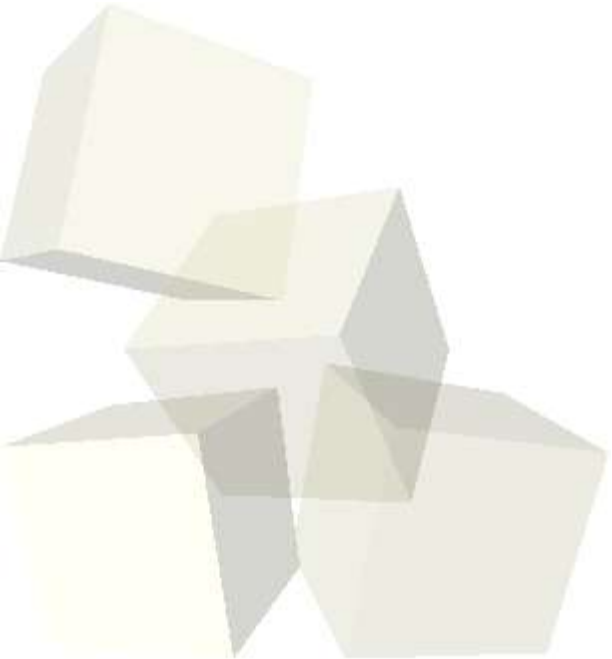
- Do you have any questions about the syllabus or what we will be doing in this course?
- What are the basic structures/mechanisms you use in imperative programming languages to solve problems?





Learn By Doing

- A lot of our time in this course is going to be spent with you working on solving some example problems during class.
- I'd like each of you to log in and bring up a terminal window. We'll be doing imperative programming today so you should write stuff in C/C++.



Arithmetic

- The term computer comes from the fact that these are devices that can do math. Most programming languages allow you to type in basic math formulas into code in much the same way that you would write them on paper.
- Let's start off by having you write a little program that will solve a quadratic equation. For the first cut we can hard code in a, b, and c. If you remember how we can also make it so it reads those values in.

Conditional Statements

- Of course, to make any type of useful program, we have to be able to specify that some sections of code should happen in certain situations and not in others. This is called conditional code and is supported by the “if” statement in most languages.
- The quadratic program you just wrote has a bit of a problem. Try making a=b=c=1. What happens in that case? What should happen?
- Add some conditional code to your program to make it do the right thing in this case.

Boolean Logic

- In order to make conditional statements work we have to have boolean expressions. These are expressions that evaluate to either true or false. They tell us whether we should be doing something or not.
- Simple boolean expressions can be made by doing comparisons between numeric values.
- To make more complex boolean expressions we use boolean operators like and, or, not, and xor.
- How do you do each of these in the language that you are using?

Loops

- Programs also typically have a need to repeat certain operations more than once. This is done with looping structures.
- Loops typically are denoted with the words for or while, though exactly what those things mean can vary significantly from one language to the next.
- To illustrate loops we will write programs that explore two different fractals.
  - The Mandelbrot set is defined as the set of points, C, in the complex plane where the following sequence stays finite: $z_{n+1}=z_n^2+C$, $z_0=C$.
  - Serpenski's triangle can be made by taking 3 points in the plane, then picking a 4^{th} point at random. Pick on of the 3 points randomly and move halfway to it.

Minute Essay

- How comfortable did you feel doing the things that we did today? What things, if any, caused issues for you?
- Do you feel comfortable with the concepts of conditionals and iteration even if their practice in a particular language is challenging?
- Feel free to play with the examples we looked at today or other similar examples to re-familiarize yourself with these ideas.