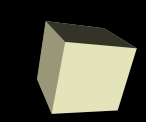
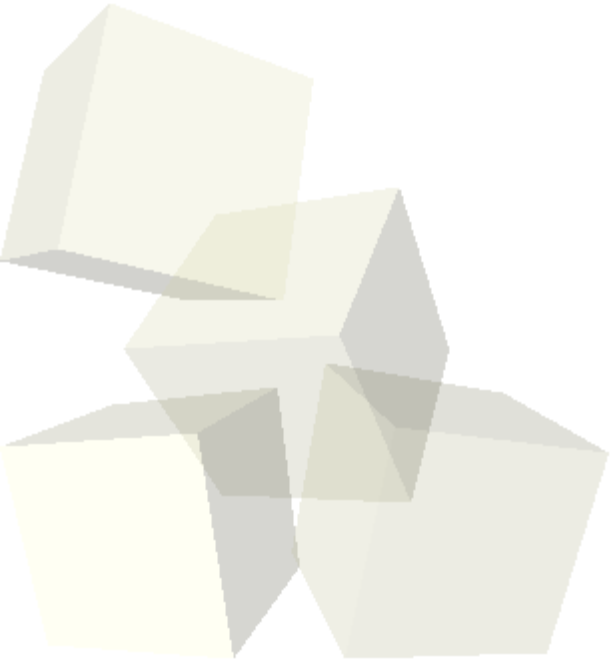
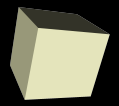


Machine Arithmetic

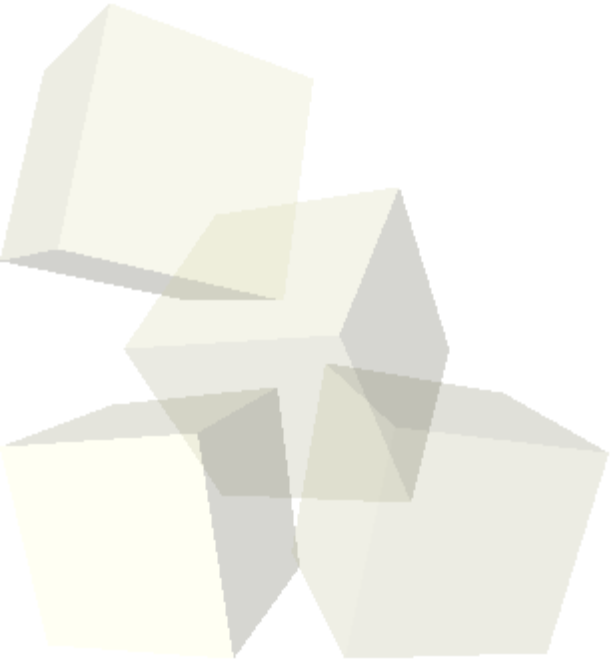
8/31/2007

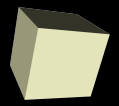




Opening Discussion

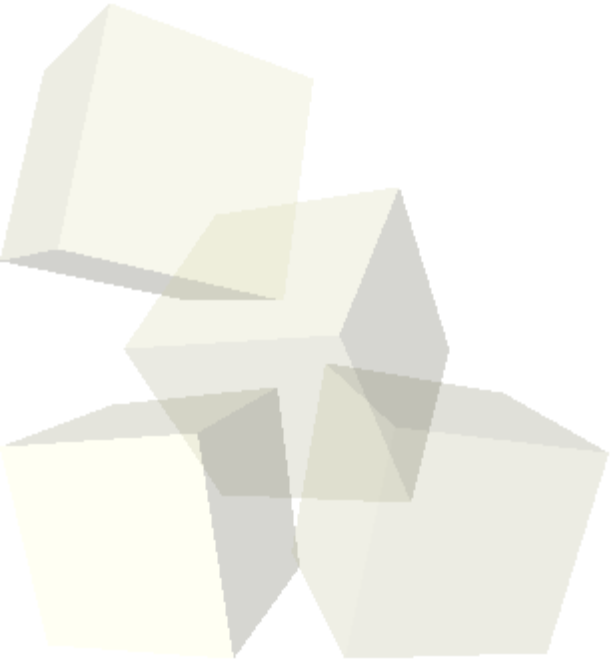
- Let's look at some interclass problems.
- If you played with your program some you probably found that it behaves oddly in some regards. Why is this?
- What did we talk about last class?





Output with printf

- We have seen printf, but we haven't really used the f part of it.
- Let's look at the man page for printf and see if we can figure out what it is saying.
- Let's play with printf some. We can also use the sizeof operator to learn a bit more about the primitive types on our system.



Input with scanf

- The opposite of printf is scanf. Use this function when you want to read input from the user.
- Let's look at the man page for scanf.
- At this point, all the variables you pass to scanf will need to have & in front of them. This is because scanf needs to know where the variable is in memory and that is what & does. We'll go into more detail about this when we cover pointers.

I/O Redirection in Linux

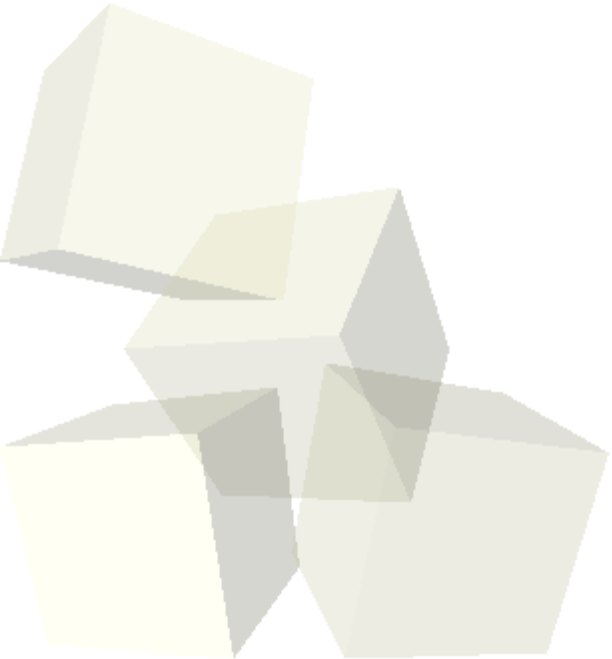
- There is another feature of the command line interface that provides tremendous power that we haven't discussed yet: I/O redirection.
- Not only does this make the command line more powerful, it will be really helpful on some assignments.
- You can make a program use a file as standard input or standard output.
  - command < input > output
  - You don't have to specify both.
- The output of one program can be “piped” so become the input of another program with |.

Numbers on Machines

- We've already said that computers use binary.
- 8 binary digits, bits, are called a byte.
- The machine's native size chunk of memory is called a “word”.
- These machines have a 32-bit word. If you have a 64-bit machine it will use a 64-bit word.
- We can use the sizeof operator in C to look at the sizes of the different primitive types. What it reports is a multiple of the size of a char which is normally a single byte.

Binary Addition

- Adding binary numbers is quite simple. You just do a lot of carrying.
- Let's do some examples. We'll assume we are working with 8-bit numbers. The number of bits we have can matter.



Twos Complement and Subtraction

- Subtraction is just addition of the negative. So how do we make negatives in binary? Remember, you don't have a -. You only have 0 and 1.
- There are actually three different ways that negatives have been done in binary. Your book covers all three, but I will only discuss the one that computers today use.
- Twos compliment is based on the idea that a+(-a)=0. So we define -a to be the number such that when added to a we get zero.
- This is possible because we have a finite number of bits. Your book discusses an easy shortcut.

Bit Shifting

- In decimal when you multiply/divide by a factor of 10 you are just moving the decimal point around.
- In binary the operation of moving all the bits left or right is called bit shifting and it is equivalent to multiplication or division by powers of two.

Multiplication

- To multiply two binary numbers do standard long multiplication. Just remember when you are adding that you are adding binary numbers.
- Note that adding more than two binary numbers can get tricky. It can result in carrying operations you have never done by hand in decimal.

Binary Fractions

- If the digits in binary start with one and move to larger powers of two moving to the left, what happens if you put in a “binary point” and move to the right?
- Numbers that have a fixed number of bits with a point at a selected position are called fixed point numbers. These were popular before floating point accelerators became common, but now they are only used on specialty hardware with low end chips.

Floating Point Numbers

- Numbers that can have fractional values are represented with floating point numbers on virtually all modern machines.
- Floating point numbers are basically like scientific notation in binary.
  - $v=(-1)^{s} * (1+frac) * 2^{(exp-bias)}$
- Single precision numbers use one bit for the sign, 8 for the exponent, 23 for the fractional part, and have a bias of 127.
- Double precision uses one bit for sign, 11 for exponent, 52 for fraction, and has a bias of 1023.

Limitations of FP

- Floating point numbers are not actually the real numbers you know from math even though we often use them that way.
- They have limitations because of the fixed number of bits.
- Some of the math properties you are used to for numbers don't actually hold for floating point numbers.
- Example: (a+b)+c=a+(b+c)
  - Try this with a, b, and c as floats and make a=1e7, b=-1e7, c=0.01.
- You also have to be careful of things like subtracting two numbers that are almost equal.

Minute Essay

- Give me the 8-bit representation of -5.
- Interclass Problem – Write a program to demonstrate potential problems with floating point numbers. I suggest using floats instead of doubles to make it easier.
- There is a quiz on Wednesday after the long weekend. It can cover anything that we have talked about in class or in the readings.