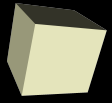




11/5/2007





# Opening Discussion

- Let's look at solutions to the interclass problem.





# Arrays of Strings

- Some of you have already dealt with arrays of strings, and we worked some last class on making a function that could handle arrays of strings.
- We can treat an array of strings either as a 2-D array of characters, an array of pointers, or a pointer to a pointer. Which we choose depends mainly on the flexibility we want. The first is the easiest to code, but the least flexible.
- Using any of the methods with pointers will inevitably require you to use malloc and free for dealing with dynamic memory.



# Command Line Arguments

- As you have seen in Linux, programs can be given extra information by typing extra words after the execution command. We can make our C program do this too.
- To make a program accept command line arguments we simply write our main so that it takes two arguments instead of void.
  - ◆ `int main(int argc, char **argv)`
- The first is how many arguments are supplied, including the command for running the program.
- The second is an array of strings listing all the arguments on the command line.
- Wildcards are expanded.



# Parsing Functions

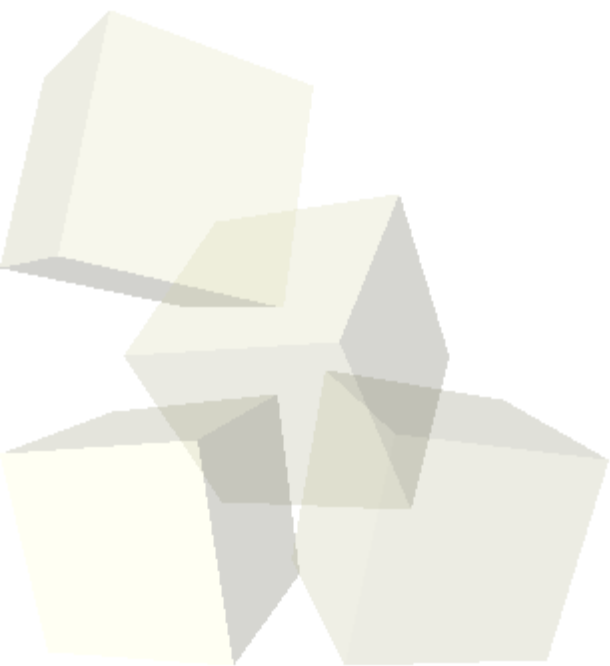
- Let's look at the following functions to see how they work.
  - ◆ strchr
  - ◆ strstr
  - ◆ strspn and strcspn
  - ◆ strtok
  - ◆ strtol and strtod
- These functions can come in very handy when you are trying to parse through data in the form of strings.





# sscanf and sprintf

- These helpful little functions work like scanf and printf except that they go into or pull out from a string instead of a stream.
- Let's look at the man pages.





- Given a string that contains a number, what are three ways you could get the numeric value using library functions?
- Interclass Problem – Do problem 36 on page 741.

