



# Basic Loops

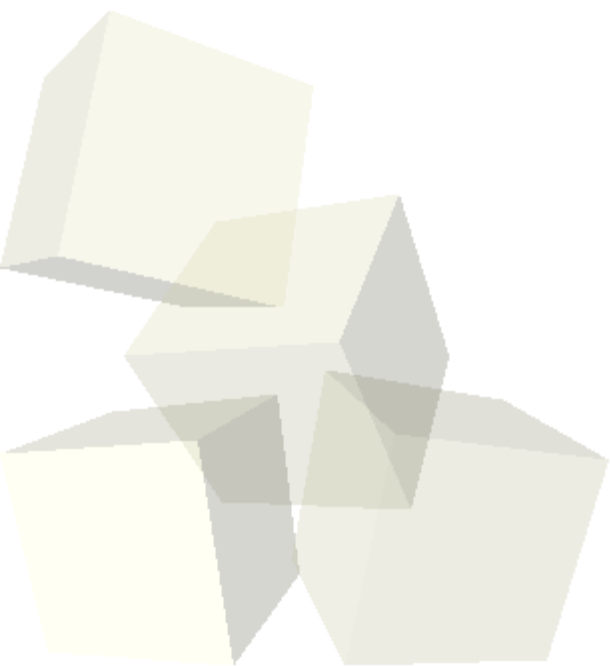
9/21/2007





# Opening Discussion

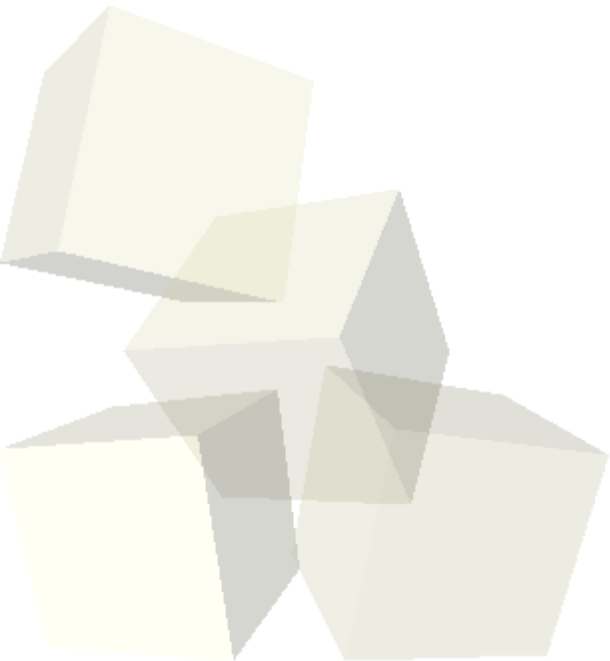
- Let's look at solutions to the interclass problem.
- Submitting assignment code.
- What happens when you leave out a break in a switch statement?





# Finishing the switch Code

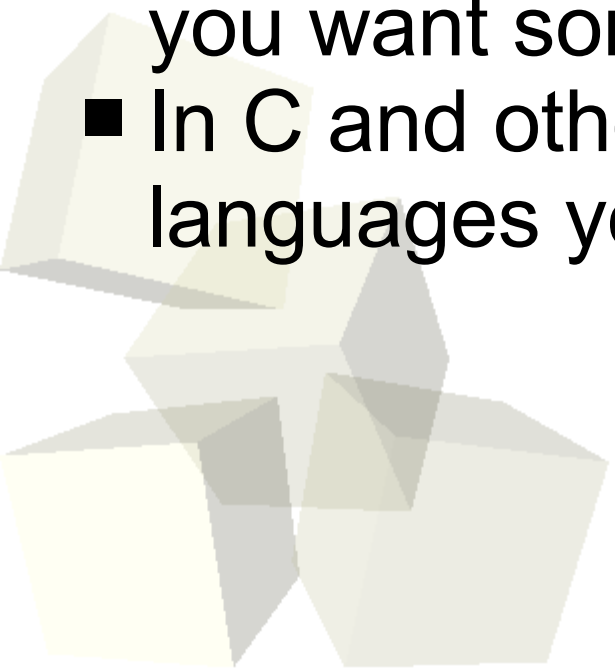
- Let's take a few minutes to finish off the code that we started for our calculator last time.





# Repetition

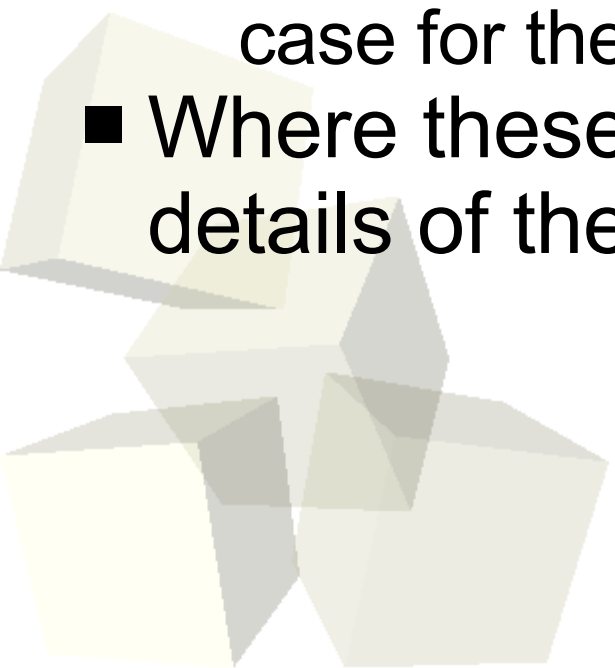
- Making things happen more than once is a critical aspect of computing.
- Last class we saw how we can get an action to happen multiple times with recursion.
- Doing basic repetition with recursion isn't ideal in C because of stack overflow problems and it takes a lot more coding to write a new function anytime you want something to happen more than once.
- In C and other structured or object-oriented languages you typically do repetition with a loop.





# Parts of a Loop

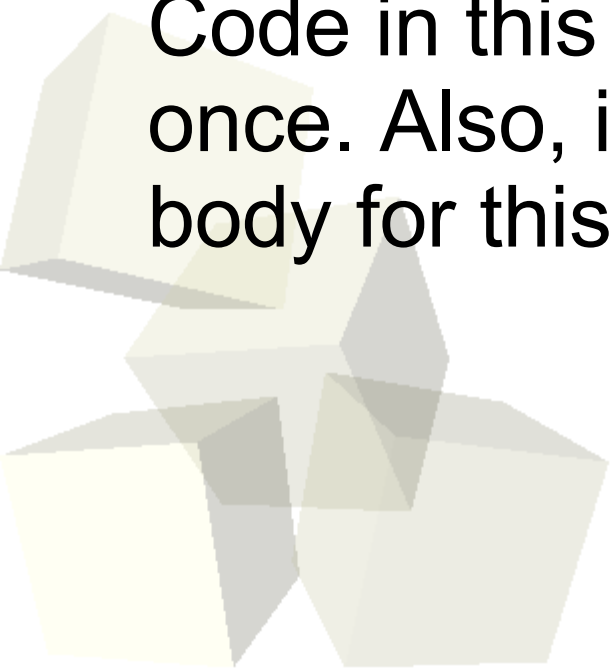
- All loops have four basic parts, though some can be implicit in certain cases.
  - ◆ Initializer – sets up variables before the loop starts.
  - ◆ Conditional – this determines whether the loop will execute.
  - ◆ Body – the statements that do the work you want done in the loop.
  - ◆ Iterator – something that moves you on to the next case for the loop.
- Where these pieces appear depends on the details of the loop.





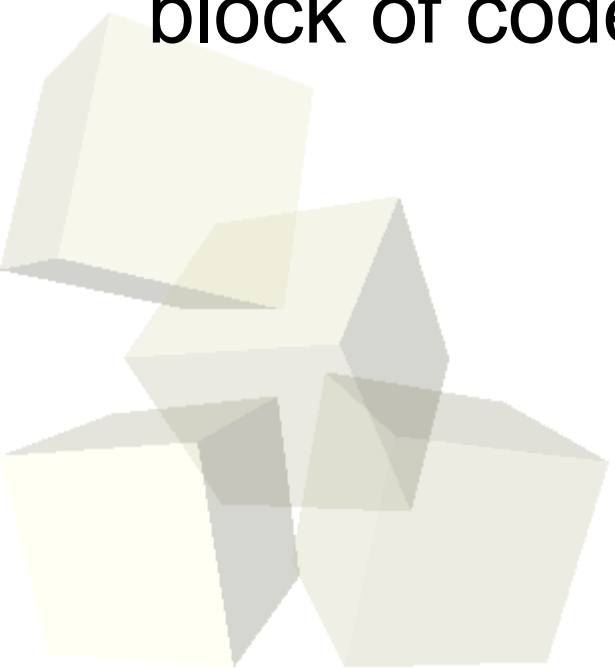
# Pre-test vs. Post-test

- One very significant distinction between loops is that of pre-test vs. post-test.
- A pre-test loop tests the condition before it executes the first time. As a result, the body of a pre-test loop might never happen.
- A post-test loop checks the condition the first time after the body and iterator have executed once. Code in this type of loop always happens at least once. Also, initialization can happen in the loop body for this type of loop.



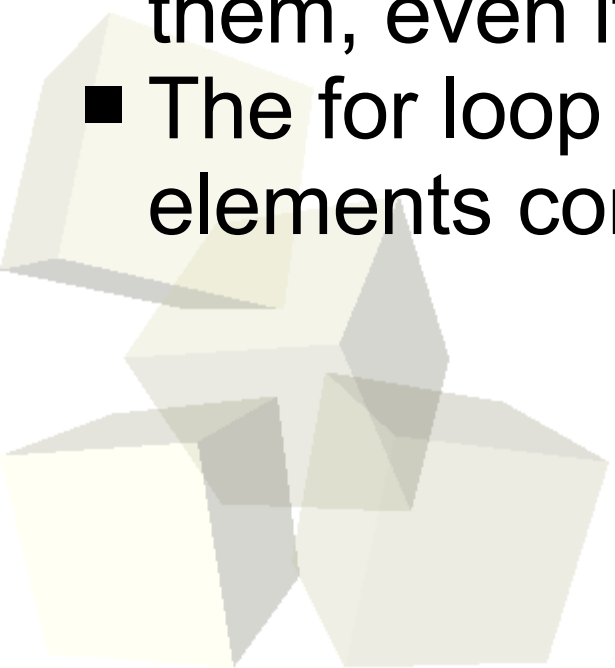


- The while loop is a simple pre-test loop.
  - ◆ `while(condition)` statement
- The initialization needs to happen before the loop and something in the body or the condition needs to play the roll of iterator. Failure to iterate a while loop is the most common form of infinite loop.
- With loops it is rare to not use curly braces and a block of code.





- This is probably the most commonly used loop in C because it brings all the elements together.
  - ◆ `for(init; condition; iterator) statement`
- for loops are very handy for counting, but notice they are really while loops with extra decoration and can be used for any purpose.
- Note the semicolons. You have to have two of them, even if you leave out one of the elements.
- The for loop makes it harder to forget different elements compared to the while loop.







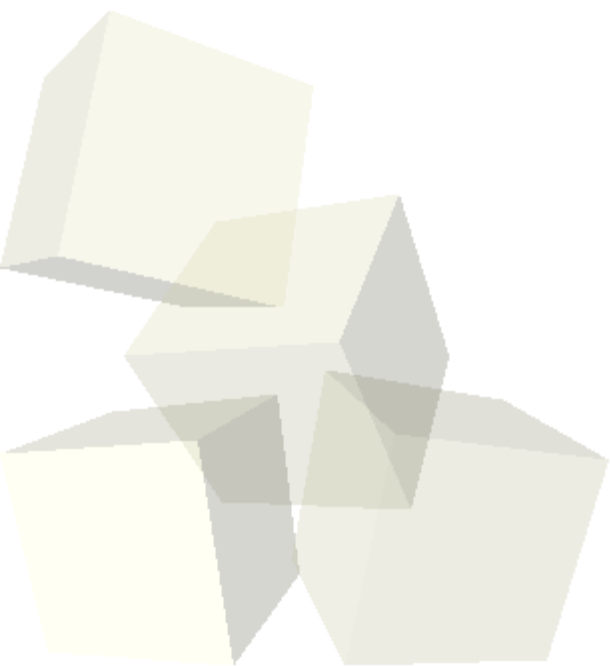
# do-while Loop

- This is the one post-test loop in C.
  - ◆ `do statement while(condition);`
- This is the least frequently used loop, but it can be quite handy when the contents of a loop should always happen once and the body of the loop will implicitly do initialization.





- Let's play with some loops. If we have time we can make our calculator use a loop instead of recursion.





# Minute Essay

- Write a for loop that prints even numbers between 0 and 10.
- Remember that assignment #1 is due tonight by midnight.
- Interclass Problem – Write a program that reads a bunch of numbers from the user. It should keep reading until the user types in a negative number.

