Recursion and Testing

9-26-2001

Opening Discussion

- What did we talk about last class?
- In the last two classes we covered all of the conditional structures in C++. Now we are going to turn to some applications.
- The first application is recursion. From the reading, what is recursion?
- We will also talk some about testing. How do conditionals effect the testing that we need to perform on a program or piece of code?

Recursion

- A recursive function is a function that calls itself. While this is a very simple and straightforward definition, it is a remarkably powerful concept in computer programming and computer science in general. In some languages recursion is the accepted method of getting code to execute multiple times.
- Recursion requires conditionals because you have to be able to return without recursing for the program to ever exit.

A Simple Loop with Recursion

- One thing you want to be able to do in a program is have a section of code execute a variable number of times depending on the inputs to the program.
- Recursion can do this by having a function call itself when it has finished. It does this repeatedly until some condition is met.
- Let's look at a simple example of this.

Recursion in the Assignment

- In the second assignment you are asked to go through the process of reading in grade data for multiple students using recursion.
- This means you will want a function that reads in the data (or calls other functions to read in the data) which calls itself repeatedly until the first name it reads in is "quit". (firstName=="quit")

More Complex Recursion

- The use of recursion is more significant when a function calls itself multiple times.
- In this case it is not just a fancy way of doing a loop, it uses the natural stack structures of function calls to help simplify processes that aren't linear in nature.
- In this class we will use recursion to help break problems into parts. It can also be useful for walking through "trees".

Testing with Conditionals

- It is impossible to prove that a program is correct in general. That doesn't mean we shouldn't try to make them correct. Instead we try testing them in specific ways.
  - All branches - Whenever there is a branch in the program you want to go down all possible branches.
  - All paths - This is the impossible goal of going down all combination of branches.

Sample Code

- Now lets take a second to look at a sample program that uses recursion to see how it works and what we need to do to test it.

Minute Essay

- What did we talk about today? This is our last day of talking about conditionals as a main topic. Next class we will begin discussing loops. Are there any topics that you have questions about that you would like for me to address briefly on Friday that deal with conditionals?
- Remember to start reading chapter 5 on loops and you should probably be working on assignment 2 before Friday.