Boolean Expressions and if

9/14/2007

Opening Discussion

- Do you have any questions about the quiz?
- Let's look at solutions to the interclass problem.
- Minute essay questions.
  - What functions will we be using?
  - When do we get to recursive functions?
  - Can you put functions in functions?
  - What happens if you are thrown into the fountain when there is no water?
  - What happens to the return value?
  - Making libraries and our own include files. How do we include .c files?
  - Using a function more than once in a program.
  - Function calls are just expressions. You can put them anywhere you use an expression.

More Minute Essays

- Finding all possible functions.
- What can we do other than math?
- Functions in operating systems?
- Relationship between called functions and calling functions.
- Getting the size of an array in C.
- SQL server software.
- Functions to do derivatives and integrals of 3-D functions?
- What functions might help in writing a game?
- What functions besides goto are really bad to use? (Goto is not a function.)

Conditional Execution

- So far when we have run a program, every line in the program executes in the same order every time. Functions allow us to break it up and user input allows different results, but everything happens the same way each time.
- To really get more control we need to be able to select what lines execute in different situations. This is called conditional execution.
- Like most languages, C has several constructs that allow us to do conditional execution.

Boolean Logic

- Before we get into the conditional constructs we need to cover some concepts in logic.
- Boolean logic is logic that involves expressions that are either true or false.
- In C, Boolean expressions are actually numeric expressions where we interpret 0 to be false and everything else to be true.
- The fact that Boolean are just ints can lead to some interesting bugs in C.

Boolean Expressions

- There are a number of operators in C that are intended for building up Boolean expressions. These are expressions that should be interpreted as either true or false.
- The comparative operators are binary operators that do numerical comparisons. We use == to check equality and != for inequality. The others are self explanatory. **WARNING**: = vs. ==.
- The explicitly Boolean operators are as follows.
  - ! - not
  - && - short-circuit and
  - || - short-circuit or (not exclusive)
- Use of parentheses is recommended.

if Statements

- The most basic conditional statement and the most obvious use of Boolean expressions is the if statement.
- Basic syntax
  - if(expression) statement
- Syntax with else
  - if(expression) statement else statement
- Remember that a block us code is a type of statement in C. Many programmers will always use a block of code with an if, but technically it takes just a statement.

Example Code

- For my roller skating class I'm adding a component of the grade based on an endurance test where you have to skate for 12 minutes. This component is worth 20 points. The grade you get is 0 for 20 or fewer laps and 20 for 40 or more laps. Between those extremes you get one point for every lap over 20.
- Let's write a function that takes the number of laps and returns the number of points you get.

Minute Essay

- What is the significance of short-circuit operators?
- Interclass Problem – Write a program that includes a comment involving at least two comparisons and two Boolean operators. Include a comment describing what your conditional is testing.