Boolean Expressions and if

9-14-2006
■ Do you have any questions about the quiz?
■ What did we talk about last class? What code did we write last class? What problems did we run into with that code?
■ Do you have any questions about the reading?
Last time we ran into a situation where we wanted to do one thing if a variable was positive and something else if it was negative. Let's go look at that code.

The way we typically deal with situations like this in C is to use an if statement.

What does an if statement look like in C? How can we use them?

Let's add the proper if statement to the code in our program.
Boolean Expressions

- The expression in an if statement is interpreted as a boolean expression. This is an expression that has only one of two values, true or false.
- In C99 we could use a variable of type bool in a boolean. In C90 you can use any numeric value and 0 will be taken as false while any other value will be true.
- Most of the time though we want to use real boolean expressions. These require operators that return true or false depending on their arguments.
We can make simple boolean expressions by comparing expressions of other types. We do this with the following comparison operators.

- `==` tests equality
- `!=` tests inequality
- `<, >` are less than and greater than
- `<=, >=` are greater than or equal to and less than or equal to

The results of these on numeric types should be clear. For character types what is compared is the ASCII value of the character.
We can combine boolean expressions into more complex boolean expressions using boolean operators.

What are the meanings of the following boolean operators?

- `&&`
- `||`
- `!`

What are the precedence of these operators?

When would you use these? Let's put a bit more logic into our program so we can see the use of this.
The problem with an `if` statement is that it doesn't have a value. As a result, some things are easier to do with the conditional expression in C. This has the form `(expr)?(true val):(false val)`. You probably shouldn't use this too much in your programming as it isn't widely used and might confuse some people.

It is standard in C and all the C-family languages so you should know it and use it when it significantly simplifies your code.

Could we have used this in our code? Would it have been beneficial?
If you do your own taxes you probably realize that the tax code includes some significant boolean logic. Assuming appropriate variables, write the boolean expression you would use for a situation where someone must be under 25 and have a gross income between $20,000 and $50,000.