

More on Conditionals

9-23-2002

Opening Discussion

- Do you have any questions about the quiz?
- What did we talk about last class?
- Last minute essay answer

```
if(age<30) {  
    rate=age*0.1;  
}  
if(age>20 && age <50) {  
    rate=rate+age*0.05;  
}
```

Nested if/else Statements

- In C, you can have all types of nesting of structures. Expressions contain other expressions and the statements in an if block can include other if blocks.
- This can happen arbitrarily deep, but beyond a certain point you have to worry about the understandability of your code.
- There are a number of times when and if statement will be the only one in an else block.

Tertiary Operator

- Earlier I mentioned an operator that took three arguments. The syntax is like this:

```
(condition)?(val1):(val2)
/* parenthesis not required */
```

- This is basically like a tiny if/else statement in one expression. The first argument is a conditional, the second is the value of the expression if it is true and the third is the value if it is false.

The switch Statement

- The other conditional statement in C is the switch statement. This is useful when you are selecting between several possible values of an integer expression. Keep in mind that characters are integers.

```
switch(intExpression) {
    case m: statements; break;
    case n: statements; break;
    default:
}
```

The Role of break

- The code in a case doesn't have to be all on one line. The break keyword tells the program to jump to the end of the block (where the '}' is).
- If a case doesn't end with a break or a return (more on that later) execution will "fall through" to the next case. This needs to be well commented because it is a common source of errors.

Debugging and Types of Bugs

- No matter how hard you try, you are very likely to write errors into your code. These are called bugs for historical reasons. Unlike the first bug though, yours can all be blamed on you.
 - Syntax errors - Simplest to find because the compiler tells you. Leaving out a ';' or a ')' are examples.
 - Runtime Errors - The program compiles, but crashes when you run it. Typically the crash happens where the error is and a debugger can give you a line.
 - Logic Errors - Programs compiles and runs, but the output is not correct. It's completely up to you to figure out what went wrong.

A Side Note #define

- We have see the #include preprocessor directive, now we will introduce another one, the #define directive.
- Code matching the define is replaced with what follows it before compile.

```
#define CONSTANT 10  
#define MIN(x,y) ((x)<(y))?(x):(y)
```

Example Code

- For our code today we will write a very simple one operation at a time calculator. How do you think that we should go about doing this?

Minute Essay

- In what situations might you choose to use a switch statement instead of if/else statements?
- Read Chapter 6 for next class.
- I will be posting the description of assignment #3 before next class. Remember that if you feel you need practice with programming you can always to the book exercises.
