



- ✓ If you use one equal sign rather than two, you don't get an error; however, the program is wrong. The nearby Technical Stuff sidebar attempts to explain why.
- ✓ If you have programmed in other computer languages, keep in mind that the C language has no `2ewd` or `fi` word. The final curly brace signals to the compiler that the `if` statement has ended.
- ✓ Also, no `then` word is used with `if`, as in the `if-then` thing they have in the BASIC or Pascal programming language.

## *A question of formatting the `if` statement*



The `if` statement is your first “complex” C language statement. The C language has many more, but `if` is the first and possibly the most popular, though I doubt that a popularity contest for programming language words has ever been held (and, then again, `if` would be great as Miss Congeniality but definitely come up a little thin in the swimsuit competition).

Though you probably have seen the `if` statement used only with curly braces, it can also be displayed as a traditional C language statement. For example, consider the following — one of the modifications from the GENIE1 program:

```
if(number==5)
{
    printf("That number is 5!\n");
}
```

In C, it's perfectly legitimate to write this as a more traditional type of statement. To wit:

```
if(number==5) printf("That number is 5!\n");
```

This line looks more like a C language statement. It ends in a semicolon. Everything still works the same; if the value of the number variable is equal to 5, the `printf()` statement is executed. If `number` doesn't equal 5, the rest of the statement is skipped.

Although all this is legal and you aren't shunned in the C programming community for using it, I recommend using curly braces with your `if` statements until you feel comfortable reading the C language.