is valid and will work properly. However, if you happen to mistype

```cpp
if (3 = myNumber)
```

the compiler will generate an error message, as it believes you are attempting to assign a value to a literal (3 always equals 3, and can't be assigned another value). Suppose you made a similar mistake made using the former notation:

```cpp
if (myNumber = 3)
```

The compiler will simply assign the value 3 to myNumber, and the block within the `if` will run— a very common error, and a difficult error to find. As a general rule, writing code that allows the compiler to find errors is much easier than repairing the causes of mysterious faulty results.

**Logical Expressions**

Often you must test for more than one condition. For example, for a character to be a lowercase letter, its value must be greater than or equal to 'a' and less than or equal to 'z'. Or, if you ask a user to respond with a y or an n, you want to accept uppercase (Y and N) as well as lowercase. To meet this kind of need, C++ provides three logical operators to combine or modify existing expressions. The operators are logical OR, written `||`; logical AND, written `&&`; and logical NOT, written `!`. Examine them now.

**The Logical OR Operator: `||`**

In English, the word *or* can indicate when one or both of two conditions satisfy a requirement. For example, you can go to the MegaMicro company picnic if you *or* your spouse work for MegaMicro, Inc. The C++ equivalent is the logical OR operator, written `||`. This operator combines two expressions into one. If either or both of the original expressions are `true`, or nonzero, the resulting expression has the value `true`. Otherwise