enough precision to hold a typical int value, and its range is much greater.

The Boolean function is_int() uses the two symbolic constants (INT_MAX and INT_MIN) defined in the climits file (discussed in Chapter 3, "Dealing with Data") to determine whether its argument is within the proper limits. If so, the program returns a value of true; otherwise, it returns false.

The main() program uses a while loop to reject invalid input until the user gets it right. You could make the program friendlier by displaying the int limits when the input is out of range. Once the input has been validated, the program assigns it to an int variable.

**Logical Operator Facts**

As we mentioned, the C++ logical OR and logical AND operators have a lower precedence than relational operators. That means an expression such as

\[ x > 5 \land x < 10 \]

is read this way:

\[ (x > 5) \land (x < 10) \]

The ! operator, on the other hand, has a higher precedence than any of the relational or arithmetic operators. Therefore, to negate an expression, you should enclose the expression in parentheses:

\[ ! (x > 5) \quad // \text{is it false that } x \text{ is greater than } 5 \]
\[ !x > 5 \quad // \text{is } !x \text{ greater than } 5 \]

The second expression, incidentally, is always false, for \( !x \) only can have the values true or false, which get converted to 1 or 0.

The logical AND operator has a higher precedence than the logical OR operator. Thus the expression

\[ \text{age} > 30 \land \text{age} < 45 \lor \text{weight} > 300 \]