means the following:

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

That is, one condition is that \(\text{age}\) be in the range 31 to 44, and the second condition is that \(\text{weight}\) be greater than 300. The entire expression is true if one or the other or both of these conditions are true.

You can, of course, use parentheses to tell the program the interpretation you want. For example, suppose you want to use \(\land\) to combine the condition that \(\text{age}\) be greater than 50 or \(\text{weight}\) be greater than 300 with the condition that \(\text{donation}\) be greater than 1000. You have to enclose the OR part within parentheses:

\[(\text{age} > 50 \lor \text{weight} > 300) \land \text{donation} > 1000\]

Otherwise, the compiler combines the \(\text{weight}\) condition with the \(\text{donation}\) condition instead of with the \(\text{age}\) condition.

Although the C++ operator precedence rules often make it possible to write compound comparisons without using parentheses, the simplest course of action is to use parentheses to group the tests, whether or not the parentheses are needed. It makes the code easier to read, it doesn't force someone else to look up some of the less commonly used precedence rules, and it reduces the chance of making errors because you don't quite remember the exact rule that applies.

C++ guarantees that when a program evaluates a logical expression, it evaluates it from left to right and stops evaluation as soon as it knows what the answer is. Suppose, for example, you have this condition:

\[x \neq 0 \land 1.0 / x > 100.0\]

If the first condition is false, then the whole expression must be false. That's because for this expression to be true, each individual condition must be true. Knowing the first condition is false, the program doesn't bother evaluating the second condition. That's fortunate in this example, for evaluating the second condition would result in dividing by 0, which is not in a computer's realm of possible actions.

**Alternative Representations**