### Note

*Compilers often have an option to disable short-circuit evaluation.*

you cannot get a ticket. There is no longer a need to check whether you can get off work because you don't have a ticket anyway.

C++ has a feature called ***short-circuit evaluation*** that allows the same kind of determinations in your program. For example, in an expression like `in_range = (i > 0 && i < 11);`, the program first checks to see if `i` is greater than 0. If it is not, there is no need to check any further because regardless of whether `i` is less than 11, `in_range` will be false. So the program sets `in_range` to false and goes to the next statement without evaluating the right side of the &&.

Short-circuiting also occurs with the or (`||`) operator. In the case of the or operator, the expression is short-circuited if the left side of the `||` is true because the expression will be true regardless of the right side of the `||`.

### On the Net

*C++ has another set of operators, called* ***bitwise operators,*** *which allow you to work with the actual bits within a number or character. The bitwise operators allow you to apply operations such as AND and OR to the bits which make up a number or character. Although many programmers never use the bitwise operators, there are some interesting uses for them. To learn more about the bitwise operators and to see some programs which use bitwise operations, see http://www.ProgramCPP.com. See topic 7.1.2.*

## SECTION 7.1 QUESTIONS

1. In C++, what value represents false?
2. List two relational operators.
3. Write an expression that returns the numeric equivalent of true if the value in the variable `k` is 100 or more.
4. Write any valid expression that uses a logical operator.
5. Write an expression that returns the numeric equivalent of false if the value in the variable `m` is equal to 5.
6. What is the value returned by the following expression?

```
(((2 > 3) || (5 > 4)) && !(3 <= 5))
```

### PROBLEM 7.1.1

In the blanks beside the statements in the program below, write a T or F to indicate the result of the expression. Fill in the answers beginning with the first statement and follow the program in the order the statements would be executed in a running program.

```
main()
{
 int i = 4;
 int j = 3;
 int true_false;

 true_false = (j < 4);                                    _____
 true_false = (j < 3);                                    _____
 true_false = (j < i);                                    _____
 true_false = (i < 4);                                    _____
 true_false = (j <= 4);                                   _____
 true_false = (4 > 4);                                    _____
 true_false = (i != j);                                   _____
 true_false = (i == j || i < 100);                        _____
 true_false = (i == j && i < 100);                        _____
 true_false = (i < j || true_false && j >= 3);            _____
 true_false = (!(i > 2 && j == 4));                       _____
 true_false = !1;                                         _____

 return 0;
}
```



CHAPTER 7, SECTION 2

# Selection Structures

Programs consist of statements that solve a problem or perform a task. Up to this point, you have been creating programs with ***sequence structures.*** Sequence structures execute statements one after another without changing the flow of the program. Other structures, such as the ones that make decisions, do change the flow of the program. The structures that make decisions in C++ programs are called ***selection structures.*** When a decision is made in a program, a selection structure controls the flow of the program based on the decision. In this section, you will learn how to use selection structures to make decisions in your programs. The three selection structures available in C++ are the if structure, the if/else structure, and the switch structure.

## USING if

Many programming languages include an ***if structure.*** Although the syntax varies among programming languages, the *if* keyword is usually part of every language. If you have used *if* in other programming languages, you should have