

Chapter 16 – The If Statement

Programs written in 3rd generation programming languages, such as C++, react to input by using Boolean expressions to make decisions about what to do or not do. In C++, the most basic decision making statement is the *if* statement. The *if* statement uses the result of a Boolean expression to determine if a following statement should or should not be performed. The form of the if statement is as follows:

```
if (Boolean-expression)
    statement;
```

If the Boolean expression is true (or 1), the statement following the Boolean expression is performed. If the Boolean expression is false (or 0), the statement following the Boolean expression is not performed.

In the following example program, a salesperson receives a bonus of \$100.00 if he has achieved a sales goal of \$10,000.00. Whether or not the goal has been achieved, the salesperson receives a commission 4% of sales made.

```
#include <iostream>
using namespace std;

void main() {
    double sales, commission;
    long intCommission;
    cout << "Enter the total sales: $";
    cin >> sales;

    // ***** commission is calculated *****
    commission = sales * .04;

    // ***** bonus is determined *****
    if (sales >= 10000.0)
        commission += 100.0;

    // ***** round up or down to two decimal places *****
    intCommission = (long)((commission + .005) * 100);
    commission = intCommission / 100.0;

    cout << "Commission amount: $" << commission << endl;
}
```

When the previous program is run with an input amount of \$11,010.32, a commission of \$440.413 is initially calculated. Since the sales amount is above the goal, \$100.00 is added to the commission amount to make a total of \$540.413. The commission is then rounded to \$540.41. The run appears as follows:

Enter the total sales: \$11010.32
Commission about: \$540.41

When the above program is run with an input amount of \$11,010.45, a commission of \$440.418 is initially calculated. Since the sales amount is above the goal, \$100.00 is added to the commission amount to make a total of \$540.418. The commission is then rounded to \$540.42. The run appears as follows:

Enter the total sales: \$11010.45
Commission about: \$540.42

When the above program is run with an input amount of \$9,679.55, a commission of \$387.182 is initially calculated. Since the sales amount is below the goal, \$100.00 is not added to the commission. The commission is then rounded to \$387.18. The run appears as follows:

Enter the total sales: \$9679.55
Commission about: \$387.18

The statement following an if statement can be a compound statement. This allows an *if* statement to control the performance of multiple statements. The following program produces the square root of a number input by the program user. Since square roots of negative numbers are undefined for reals (floating point types in C++), the program is designed to perform the calculation only if it is possible.

```
#include <iostream>
#include <cmath>
using namespace std;

void main() {
    double num, sroot;
    cout << "Enter a number: ";
    cin >> num;
    if (num >= 0) {
        sroot = sqrt(num);
        cout << "The square root of " << num
            << " is " << sroot << endl;
    }
}
```

When run, if the user enters 45, the program returns 6.7082. The run appears as follows:

```
Enter a number: 45
The square root of 45 is 6.7082
```

When run, if the user enters -3, the program returns nothing. The run appears as follows:

```
Enter a number: 45
```

Exercises

- 1) What will the following program fragment output where:
 - a) Day is greater than Night and Clear is greater than Cloudy
 - b) Day is less than Night and Clear is less than Cloudy
 - c) Day is greater than Night and Clear is less than Cloudy
 - d) Day is less than Night and Clear is greater than Cloudy

```
if (Day > Night)
    cout << "Put on sun glasses!";
if (Clear > Cloudy && Day > Night)
    cout << "Put on sun tan lotion!";
```

- 2) A and B are integer variables. Get three sets of values for A and B in order to produce all possible outputs (i.e., one set of values for A and B should produce one of the possible outputs).

```
if (A == B)
    cout << "A and B are the same";
if (A < B)
    cout << "A is less than B";
if (A > B)
    cout << "A is greater than B";
```

Programming Assignment 16.1

Implement and use the example program of page 1.

Programming Assignment 16.2

Implement and use the example program of page 2.

Programming Assignment 16.3

Write a program consisting of a main function that outputs a message if the user enters the correct password.

Programming Assignment 16.4

Hourly workers at McFudd's are paid between \$5.50 and \$9.00 per hour worked. If a worker works more than 40 hours in a week, the works hourly pay rate for the hours past the first 40 hours are increased by 50%. This is known as "time and a half" overtime pay. For example, if a worker making \$6.00 per hour works 45 hours in one week, that worker will receive $45.0 * 6.0 + 5.0 * 6.0 * .5$, or \$295.00 in pay. Write a function that receives the hours worked and rate of pay as parameters and returns the correct amount of pay. The function should return the correct amount of pay whether or not the employee works overtime. Have the user enter the hours worked and rate of pay in a main function which then calls the function and displays the returned value.

Programming Assignment 16.5

Change the program of 16.4 so that there is no input or output in the main function. Instead, the program should contain three functions, the existing function, a function to collect information (which will require pass by reference parameters), and a function to output the actual pay. The function main should consist of calls to these three functions.

Programming Assignment 16.6

Create a series of overloaded functions named *ABS* to return the absolute value of one parameter. Create functions to return the absolute values of *int*, *long*, *float* and *double* parameters. Use these functions in a program.

Programming Assignment 16.7

Add and use a void member function *winner* to *class track* that declares a winner based on which *class car* object has moved the farthest along the track.