

# Chapter 19 – The Switch Statement

C++ contains a primitive decision statement called the *switch* statement, which tests a simple value (such as integer or character) against a various constants to determine which statement or statements to perform. The *switch* statement has the following form.

```
switch(value) {  
    case constant:  
        statement(s);  
    case constant:  
        statement(s);  
        :  
        :  
    case constant:  
        statement(s);  
    default:  
        statement(s);  
}
```

If the value matches a constant of a *case*, all sets of statements after the matching constant are performed up until the end of the case statement or until a *break* statement is encountered. (*The statement break sends the program logic to the statement after the compound statement.*) If no constant matches the value, the set of statements after the *default* option is performed. The statements can be made up of any valid C++ statement or may even be omitted.

Since each *case* is dependent on a single constant, the *switch* statement cannot be effectively used to test ranges of values. For example, the *switch* statement is an excellent choice for interpreting selections by a user from a menu, but would be impractical for the *nested-if* voting age example in Chapter 18.

The following example program uses a *switch* statement to determine a response to a letter grade input. Note the use of two *case* constants for each possible grade. This allows the user to input an upper or lower *case* letter. If the user enters a 'B', there are no statements directly following the case constant 'B'. Since there is not a *break* statement, the program proceeds to the set of statements after the *case* constant 'b'.

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

void main( ) {
    cout << " Enter your letter Grade (A, B, C, D, F): " << endl;
    char grade;
    cin >> grade;
    switch(grade){
    case 'A':
    case 'a':
        cout << "Attaboy!!!" << endl;
        break;
    case 'B':
    case 'b':
        cout << "Very Good!!!" << endl;
        break;
    case 'C':
    case 'c':
        cout << "OK!" << endl;
        break;
    case 'D':
    case 'd':
        cout << "Not so hot!!!" << endl;
        break;
    case 'F':
    case 'f':
        cout << "Surrender the car keys, now!!!" << endl;
        break;
    default:
        cout << grade << " is not a valid grade." << endl;
    }
}
```

If the user enters an 'A' when the program is run, the run will look as follows:

```
Enter your letter Grade (A, B, C, D, F): A
Attaboy!!!
```

If the user enters 'c' when the program is run, the run will look as follows:

```
Enter your letter Grade (A, B, C, D, F): c
OK!
```

If the user enters ‘8’ when the program is run, the run will look as follows:

```
Enter your letter Grade (A, B, C, D, F): 8
8 is not a valid grade.
```

The next *switch* example includes the file *conio.h* in order to incorporate the function *getch*. This function accepts a character when its key is pressed without waiting for the Enter key to be pressed. In addition, the key pressed is not reflected on the monitor screen. The program moves past the *getch* function as soon as the user presses any key. *getch* is used in the following form.

```
char_variable = getch();
```

In the following example, the user is presented with a menu. If the users presses 1, 2, 3, or 4, the program will output 1, 2, 3, or 4 beeps or rings of the system bell. (*The system bell is activated by outputting the seventh ASCII character. The char variable a\_beep is assigned the value 7 without quotes and is used to output the bell value.*)

```
#include <iostream>
#include <conio.h>
using namespace std;

void main() {
    char beeps, a_beep = 7;
    cout << "Press the digit to the left of your choice:" << endl
        << " 1 - 1 beep" << endl
        << " 2 - 2 beeps" << endl
        << " 3 - 3 beeps" << endl
        << " 4 - 4 beeps" << endl;
    beeps = getch();
    switch(beeps){
    case '4':
        cout << a_beep;
    case '3':
        cout << a_beep;
    case '2':
        cout << a_beep;
    case '1':
        cout << a_beep;
        break;
    default:
        cout << "No beeps!" << endl;
    }
}
```

If the user presses the 2 key when the program is run, the program will use the *cout* statement after *case '2'*, then will proceed to the statements after *case '1'*. Since the last statement after *case '1'* is *break*, the program bypasses the statement in the *default* section.

### **Exercises**

1. What is the purpose of the case statement in the switch statement?
2. What is the purpose of the default statement in the switch statement?
3. What is the purpose of the break statement?
4. What is the output of the following code given that  $Z = 3$ .

```
switch(Z)
{
  case 1: b = 3;
         break;
  case 2: b = 4;
         break;
  case 3: b = 5;
  case 4: b = 6 ;
         break;
  default: b = 7;
}
cout << Z;
```

### **Programming Assignment 19.1**

Implement and use the two example programs.

### **Programming Assignment 19.2**

Create and use a function that accepts a letter and returns the equivalent digit on the dial or keypad of a telephone. Telephone letters match digits as follows:

ABC = 2	DEF = 3	GHI = 4	JKL = 5
MNO = 6	PRS = 7	TUV = 8	WXY = 9

Notice that Q is not used. The function should work with upper and lower case letters.  
*Time saving hint: Adding 32 to an upper case letter changes that letter to lower case!*