The Switch Structure

If you have multiple conditions to test and multiple paths to take it could be difficult to write enough if..then statements to do everything you need.

The best way of handling this is with the **switch structure**.

The switch structure has a lot of uses but the most common one is for a menu driven system.

```
cout << "How do you want the order shipped? << endl;
cout << "1—Ground" << endl;
cout << "2—2-day Air" << endl;
cout << "3—Overnight Air" << endl;
cout << "Enter the number of the shipping method: ";
cin >> shipping_method;

switch(shipping_method)
{
     case 1:
          shipping_cost = 5.00;
          break;
     case 2:
          shipping_cost = 7.50;
          break;
     case 3:
          shipping_cost = 10.00;
          break;
     default:
          shipping_cost = 0.00;
          break;
}
```

If we look at the code above we see that the structure begins with the keyword **switch**, followed by the control expression (the variable `shipping_method`) to be compared within the structure.

Inside the structure are a series of **case** keywords. Each one provides the code that is to be executed in the even that `shipping_method` matches the value that follows the case.

The **default** keyword tells the compiler that if nothing else matches, execute the statements that follow.

The **break** keyword which appears at the end of each case segment, causes the follow of logic to jump to the first executable statement after the **switch structure**.

```
switch(character_entered)
{
     case ‘A’:
          cout << “The character entered was A, as in
          arithmetic.” << endl;
          break;
     case ‘B’:
          cout << “The character entered was B, as in
          binomial.” << endl;
          break;
     default:
          cout << “Illegal entry!!” << endl;
          break;
}
```