Visual C# selection with switch

The switch condition acts in a similar way to if but has a slightly different syntax. Switch only works for integers and characters. It cannot compare words or fancy numbers. It can be a clearer way to make decisions than multiple if or else if clauses.

This is as simple as switch gets:

```
int numberToTest =2;
switch (numberToTest)
{
        case 1:
            Console.WriteLine("1 entered");
            break;
        default:
            Console.WriteLine("something else entered");
            break;
}
```

The default is the equivalent of else, it would be irresponsible to not have it.

Here is a slightly more detailed switch:

```
int numberToTest =2;
switch (numberToTest)
{
        case 1:
            Console.WriteLine("1 entered");
            break;
       case 2:
            Console.WriteLine("2 entered");
            break;

        default:
            Console.WriteLine("something else entered");
            break;
}
```

Note the use of the break to define the different case statements. The break is not always required and some interesting results can be achieved by omitting it.

```
int numberToTest =2;
switch (numberToTest)
{
     case 0:
      case 1:
            Console.WriteLine("0 or 1 entered");
            break;
```

```
        case 2:
             Console.WriteLine("2 entered");
             break;

         default:
             Console.WriteLine("something else entered");
             break;
}
```

Char values can also be used in a switch but char and int cannot be used at the same time as only 1 variable is assessed in the switch.

```
case 'c':
      Console.WriteLine("c entered");
      break;
```

Using switch

1. Create a menu system that assigns different keyboard output to the following keys, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’ or any other letter. If the user enters a lower case character it should be converted to upper case using toUpper.

```
String data = “a”;
data.ToUpper();// data is now ‘A’
```

2. Create a program that asks the user for a number from 1 to 10. The computer generates a random number in the same range. A switch statement will output how far away the user’s data is from that selected by the computer. Both numbers (the user’s and that of the computer) should be output at the end of the program. The following code will create a random number in the range of 1 to 10.

```
Random r = new Random();
// a random number from 0 to 9.99999999
double dRand = r.Next(10);
 //cast to int and change from 0 to 9 - 1 to 10
int myNum = (int)dRand + 1;
```

If negative numbers are to be dealt with C# can find the absolute of a number as shown below.

```
int num = -2;
num = System.Math.Abs(num); // num is now 2
```