Switch Statements

This lesson discusses switch statements in C# using an example

WE'LL COVER THE FOLLOWING

Switch Case

Typically this is required when based on different values of a particular expression, different actions need to be performed. The basic construct of a switch case looks as follows:

```
switch (expression)
{
  case constant-expression:
     statement
     jump-statement
  default:
     statement
     jump-statement
}
```

Switch Case

- In code block above the `expression` can have multiple values. Essentially:
  - **string**
  - **integer**
- `case` section with `constant-expression` can have the value as
  - constant
  - expression that results in a constant

- This decides to which **case statement** control will transfer
- The `default` section is optional and only gets executed when none of the `constant-expression` matches with the `expression`
- The `jump-statement` must be there at the end of each block to step out of the **switch** case once a particular statement section gets executed.

> **Note:** There are a number of **branching statements** or **jump-statements** available in C# such as `break`, `continue`, `goto`, `return` and `throw`.

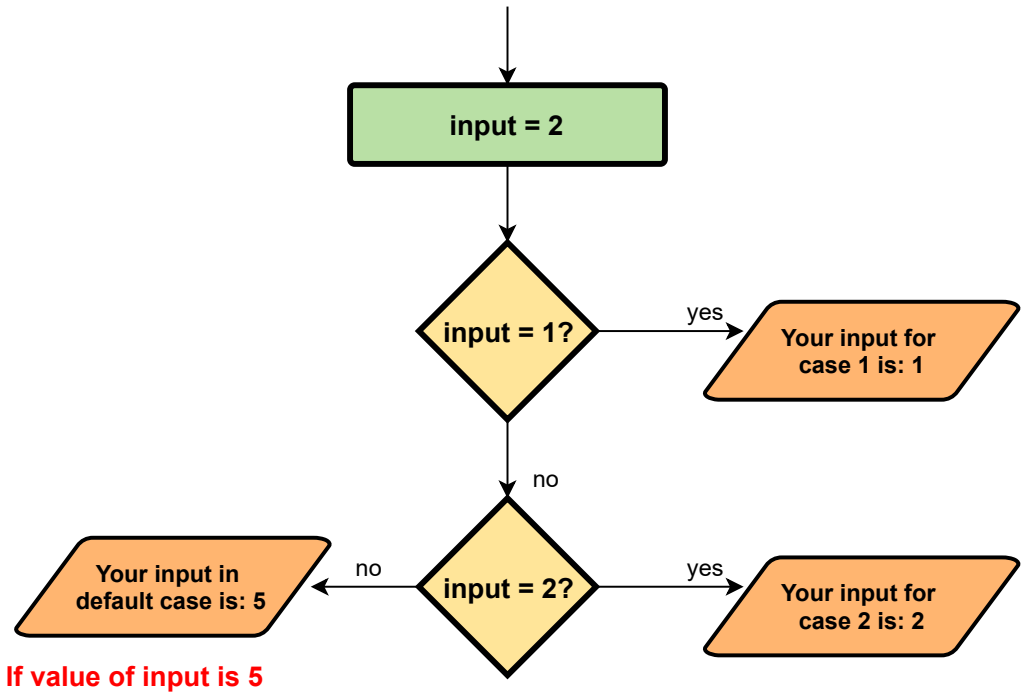
Example

Let's take a look at an example of **switch** cases to better understand the concept.

```
using System;

class SwitchExample
{
    public static void Main()
    {
        int input=2; //change value of this input to see output for different cases

        // switch with integer type
        switch (input)
        {
            case 1:
                Console.WriteLine("Your input for case 1 is: {0}", input);
                break;
            case 2:
                Console.WriteLine("Your input for case 2 is: {0}", input);
                break;
            default:
                Console.WriteLine("Your input in default case is: {0}" , input);
                break;
        }
    }
}
```



Flowchart of switch case

Code Explanation

In the code above:

- First the value of *variable* `input` is set equal to **2**.
- Then the `switch` function is called with `input` passed to it as the parameter.
- As the value of `input` is **2**, `case 2` is executed displaying: **Your input for case 2 is: 2** in console.

You can change the value of `input` in the code above to execute various **switch** cases.

- If the value of `input` is changed to **1** then switch case **1** will execute.
- If the value of `input` is changed to a number other than **1** or **2** then the `default` case will *execute*.

This marks the end of our discussion on **switch** *statements*. In the next lesson, we will discuss **ternary** *operators* in **C#**.