**C# switch:**

C# includes another decision making statement called switch. The switch statement executes the code block depending upon the resulted value of an expression.

Syntax:

switch(expression)

{

case <value1>

// code block

break;

case <value2>

// code block

break;

case <valueN>

// code block

break;

default

// code block

break;

}

As per the syntax above, switch statement contains an expression into brackets. It also includes multiple case labels, where each case represents a particular literal value. The switch cases are separated by a break keyword which stops the execution of a particular case. Also, the switch can include a default case to execute if no case value satisfies the expression.

Note : Case label in switch must be unique. It can be bool, char, string, integer, or enum.

Consider the following example of a simple switch statement.

int x = 10;

switch (x)

{

case 5:

Console.WriteLine("Value of x is 5");

break;

case 10:

Console.WriteLine("Value of x is 10");

break;

case 15:

Console.WriteLine("Value of x is 15");

break;

default:

Console.WriteLine("Unknown value");

break;

}

The switch statement can include expression or variable of any data type such as string, bool, int, enum, char etc.

string statementType = "switch";

switch (statementType)

{

case "if.else":

Console.WriteLine("if...else statement");

break;

case "ternary":

Console.WriteLine("Ternary operator");

break;

case "switch":

Console.WriteLine("switch statement");

break;

}

**Goto in switch:**

The switch case can use goto to jump over a different case.

string statementType = "switch";

switch (statementType)

{

case "DecisionMaking":

Console.Write(" is a decision making statement.");

break;

case "if.else":

Console.Write("if-else");

break;

case "ternary":

Console.Write("Ternary operator");

break;

case "switch":

Console.Write("switch statement");

goto case "DecisionMaking";

}

**Nested switch:**

Nested switch statements are allowed in C# by writing inner switch statement inside an outer switch case.

int j = 5;

switch (j)

{

case 5:

Console.WriteLine(5);

switch (j - 1)

{

case 4:

Console.WriteLine(4);

switch (j - 2)

{

case 3:

Console.WriteLine(3);

break;

}

break;

}

break;

case 10:

Console.WriteLine(10);

break;

case 15:

Console.WriteLine(15);

break;

default:

Console.WriteLine(100);

break;

}

**Points to Remember:**

1. The switch statement tests the variable against a set of constants.
2. The switch statement contains multiple case labels.
3. The switch case includes break keyword to stop the execution of switch case.
4. The default case executes when no case satisfies the expression.
5. A nested switch statement is allowed.