Switch in C++

Switch case statement evaluates a given expression and based on the evaluated value(matching a certain condition), it executes the statements associated with it. Basically, it is used to perform different actions based on different conditions(cases).

- Switch case statements follow a selection-control mechanism and allow a value to change control of execution.
- They are a substitute for long if statements that compare a variable to several integral values.
- The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.

In C++, the switch statement is used for executing one condition from multiple conditions. It is similar to an if-else-if ladder.

Switch statement consists of conditional based cases and a default case.

In a switch statement, the “case value” can be of “char” and “int” type.

Following are some of the rules while using the switch statement:

1. There can be one or N numbers of cases.

2. The values in the case must be unique.

3. Each statement of the case can have a break statement. It is optional.

Syntax:

```
switch(expression)
{
case value1:    statement_1; break;

case value2:    statement_2; break;

.....
......
......
case value_n:    statement_n; break;
```

```
default:     default statement;

}
```

Some important keywords:

1) Break: This keyword is used to stop the execution inside a switch block. It helps to terminate the switch block and break out of it.

2) Default: This keyword is used to specify the set of statements to execute if there is no case match.

> Note: Sometimes when default is not placed at the end of switch case program, we should use break statement with the default case.

Important Points About Switch Case Statements:

1) The expression provided in the switch should result in a **constant value** otherwise it would not be valid. Some valid expressions for switch case will be,

```
// Constant expressions allowed
switch(1+2+23)
switch(1*2+3%4)

// Variable expression are allowed provided
// they are assigned with fixed values
switch(a*b+c*d)
switch(a+b+c)
```

2) Duplicate case values are not allowed.

3) The **default statement is optional**. Even if the switch case statement do not have a default statement,

it would run without any problem.

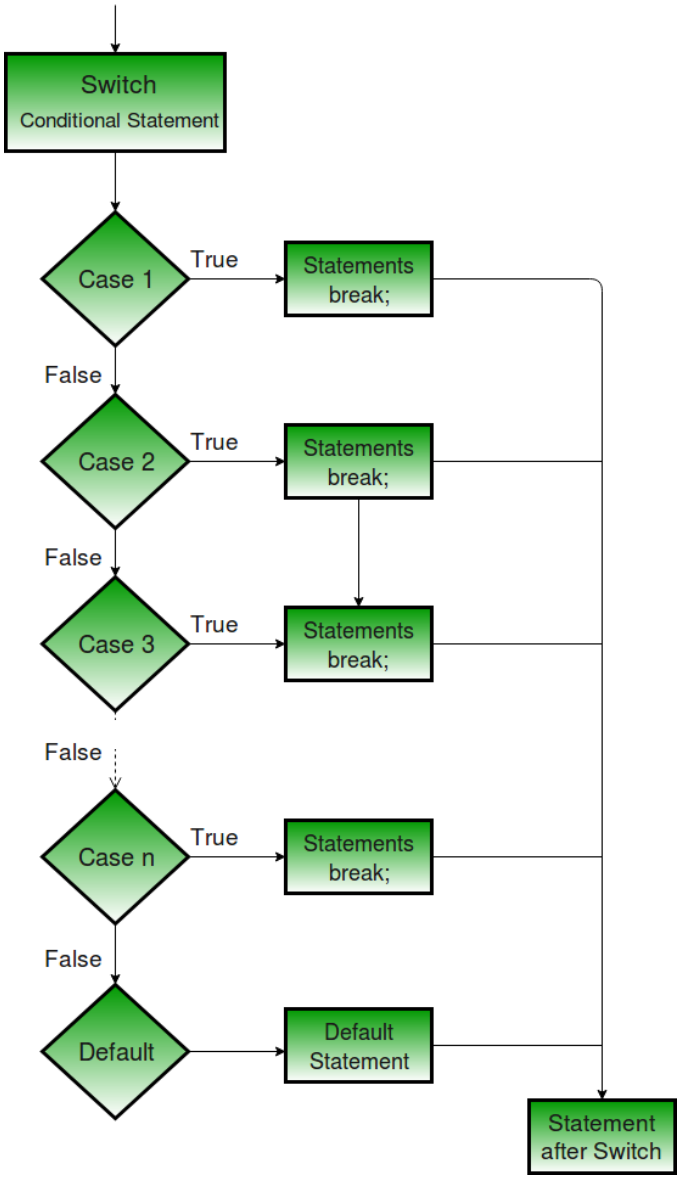
4) The **break statement is used inside the switch to terminate a statement** sequence. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.

5) The **break statement is optional**. If omitted, execution will continue on into the next case. The flow of control will fall through to subsequent cases until a break is reached.

6) Nesting of switch statements is allowed, which means you can have switch statements inside another switch. However nested switch statements should be avoided as it makes the program more complex and less readable.

7) Switch statements are **limited to integer values and characters** only in the check condition.

Flowchart:



Example:

```
// C++ program to demonstrate syntax of switch
#include <iostream>
using namespace std;
```

```
// Driver Code
int main()
{
    int x = 2;
    switch (x) {
    case 1:
        cout << "Choice is 1";
        break;
    case 2:
        cout << "Choice is 2";
        break;
    case 3:
        cout << "Choice is 3";
        break;
    default:
        cout << "Choice other than 1, 2 and 3";
        break;
    }
    return 0;
}
```

Output

```
Choice is 2
```