

C++ Switch Statement

C++ Switch

In C++, Switch statement executes one of the many code blocks based on the value of an expression.

In this tutorial, we learn the syntax of switch statement, execution flow diagram, and its usage with example C++ programs.

Syntax of Switch Statement

Following is the syntax of switch statement in C++.

```
switch (expression) {  
    case value1:  
        // code block  
        break;  
    case value2:  
        // code block  
        break;  
    default:  
        // code block  
}
```

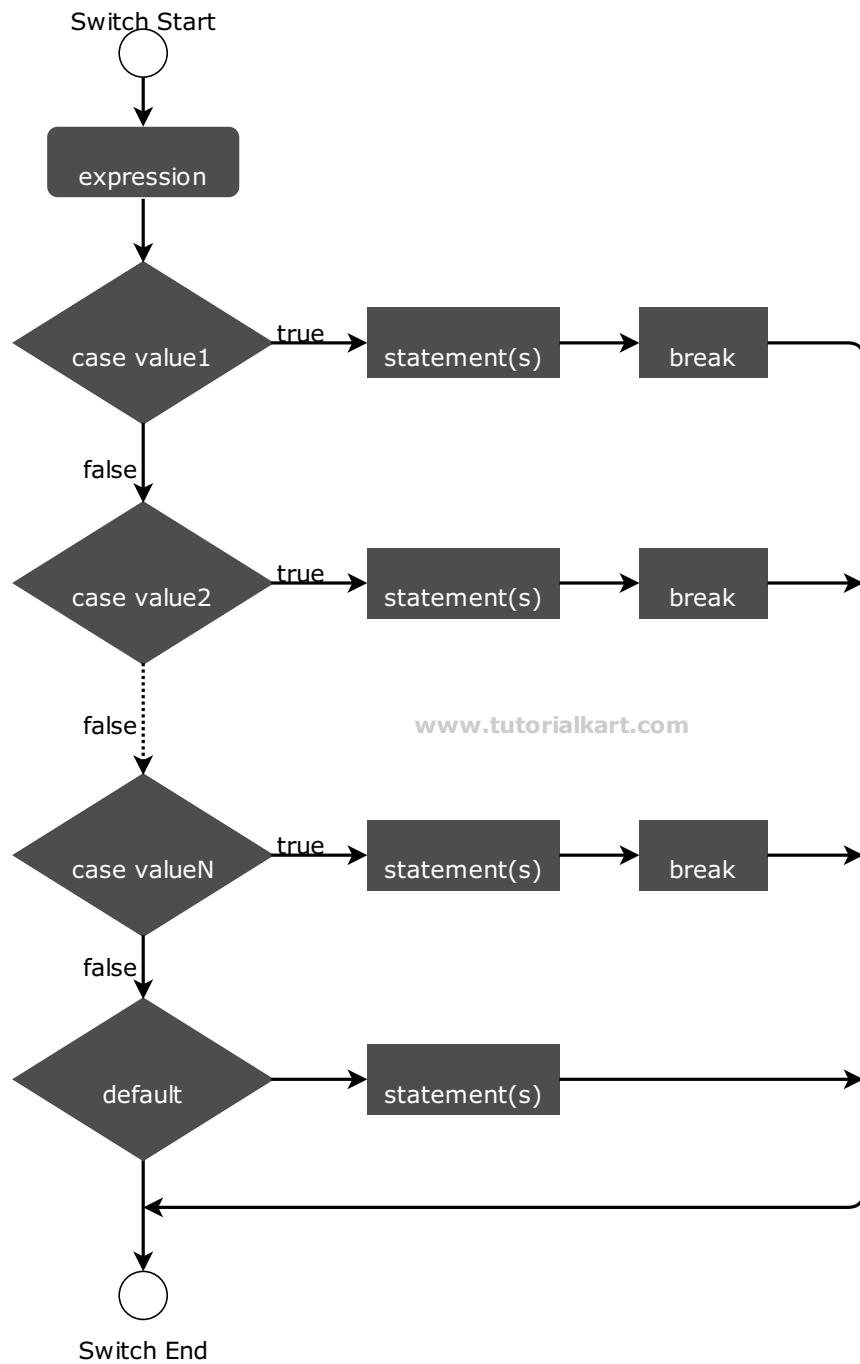
Working of Switch Statement

1. Start of switch statement. The expression is evaluated to a value.
2. This value is then compared to each `case` value.
3. If it finds a match, corresponding block is executed.
4. `break` statement at the end of case block is optional. After executing a block, execution comes out of the loop because of `break`. If no `break` statement is given, all the case blocks and default block, next to this block, are executed.
5. default block is optional. If expression value does not match any of the case values, default block is executed.

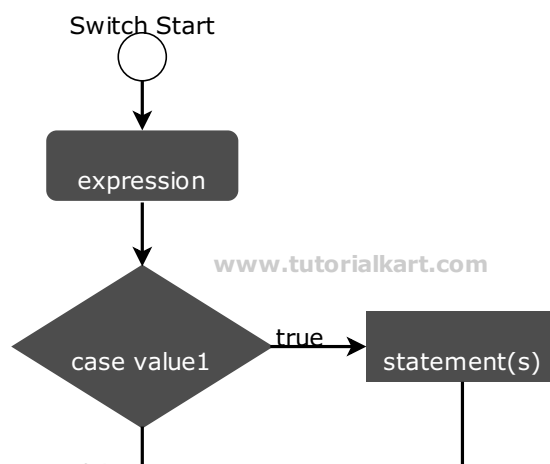
Check the following flow diagrams for detailed working of `break` statements at the end of case blocks, and default block.

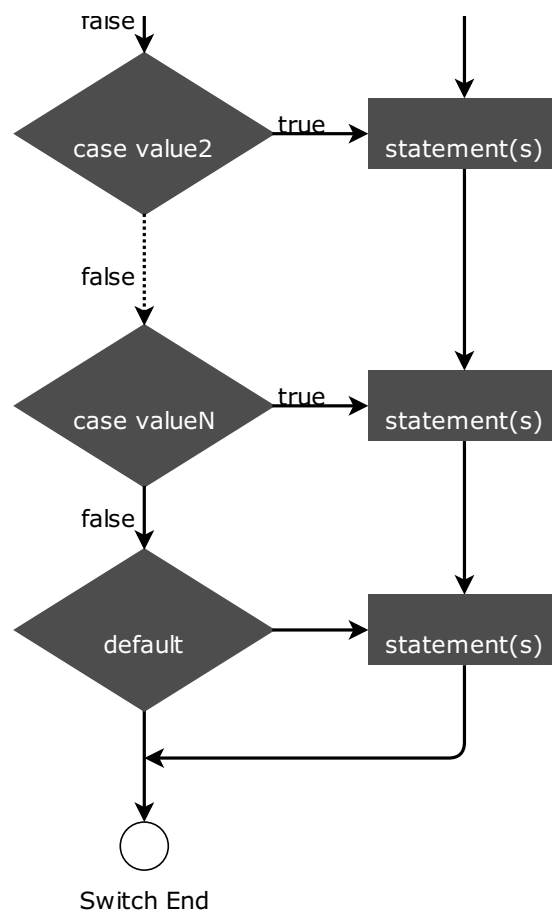
C++ Switch Flowchart or Flow-Diagram

Following is the execution flow diagram of switch statement with break statement for each of the case blocks.



Following is the execution flow diagram of switch statement without break statement for case blocks.





Example – C++ Switch

Following is an example of Switch statement in C++. In this example, we shall print if the name of weekday based on number.

C++ Program

```
#include <iostream>
using namespace std;

int main() {
    int day = 25;
    switch (day%7) {
        case 1: {
            cout << "Monday" << endl;
            break;
        }
        case 2: {
            cout << "Tuesday" << endl;
            break;
        }
        case 3: {
            cout << "Wednesday" << endl;
            break;
        }
        case 4: {
            cout << "Thursday" << endl;
            break;
        }
        case 5: {
```

```

        cout << "Friday" << endl;
        break;
    }
    default: {
        cout << "Off day" << endl;
    }
}
}

```

Output

```
Thursday
```

Example – C++ Switch without Break

Following is an example of Switch statement in C++. We have not used break statement after end of case blocks. In this example, we shall print if the name of weekday based on number.

C++ Program

```

#include <iostream>
using namespace std;

int main() {
    int day = 24;
    switch (day%7) {
        case 1: {
            cout << "Monday" << endl;
        }
        case 2: {
            cout << "Tuesday" << endl;
        }
        case 3: {
            cout << "Wednesday" << endl;
        }
        case 4: {
            cout << "Thursday" << endl;
        }
        case 5: {
            cout << "Friday" << endl;
        }
        default: {
            cout << "Off day" << endl;
        }
    }
}

```

Output

```
Wednesday
Thursday
Friday
Off day
```

Conclusion

In this [C++ Tutorial](#), we learned about C++ Switch statement, consequences of including and not including break statement after each case block, and also about default block.

C++ Tutorials

- ◆ [C++ Tutorial](#)
- ◆ [C++ Hello World Program](#)
- ◆ [C++ If Else](#)
- ⇒ **C++ Switch**
- ◆ [C++ Ternary Operator](#)
- ◆ [C++ Logical Operations](#)
- ◆ [C++ Arithmetic Operations](#)
- ◆ [C++ While Loop](#)
- ◆ [C++ Do-While Loop](#)
- ◆ [C++ For Loop](#)
- ◆ [C++ ForEach](#)
- ◆ [C++ Continue](#)
- ◆ [C++ Break](#)
- ◆ [C++ Comments](#)
- ◆ [C++ Recursion](#)
- ◆ [C++ Try Catch](#)
- ◆ [C++ String Operations](#)
- ◆ [C++ Array Operations](#)
- ◆ [C++ Vector Operations](#)
- ◆ [C++ Input Output Operations](#)
- ◆ [C++ Class](#)
- ◆ [C++ Programs](#)