

switch statement in C++

The C ++ switch statement is the same as [if-else statement](#) but the line code in if-else increases, making the program slightly complex.

[1. switch statement flow-diagram in C++](#)

[2. Example of switch statement in C++ \(with int\)](#)

[3. Example of switch statement with char data-type](#)

[4. creating a calculator using switch statement in C++](#)

[5. without break keyword switch statement](#)

To solve this problem in C++, we use the switch statement in the program. In the switch statement is the execution done step by step and every statement terminates with break keyword (except for the default statement), in which case the expression value is true, the statement becomes to execute.

It has a statement default statement which executes when the expression value of all the statements is false.

syntax

```
switch (input)
{
    case <1-exp>: first case body; break;

    case <2-exp>: second case body; break;

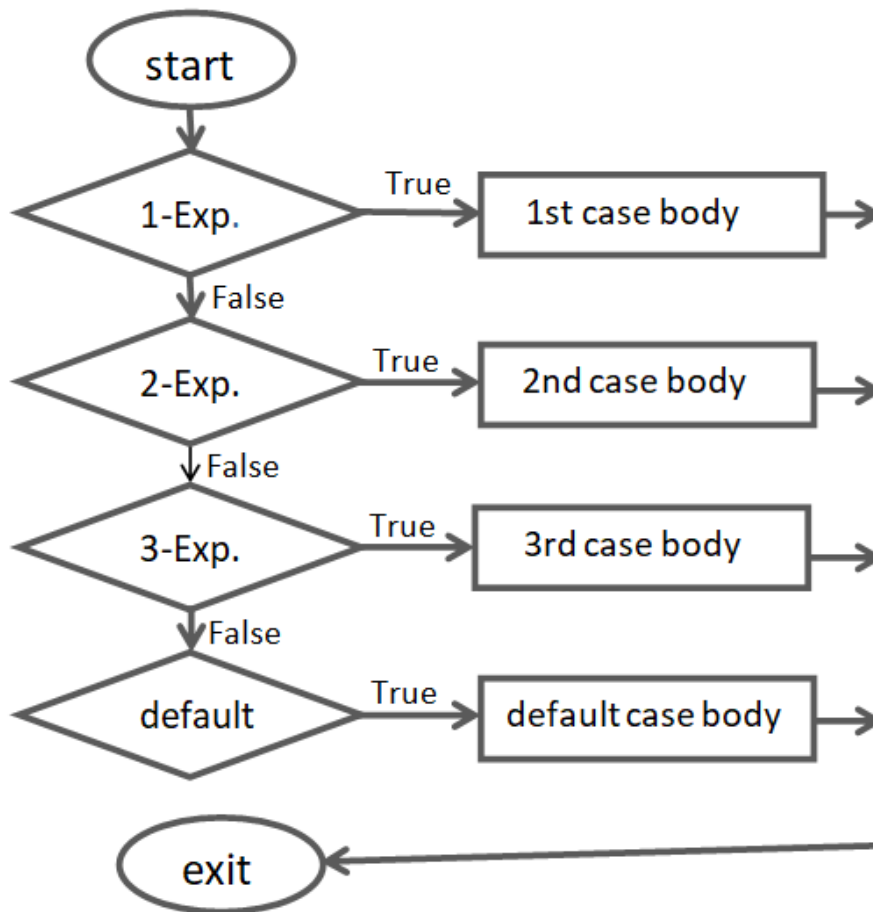
    case <3-exp>: third case body; break;

    default: body of default;

};
```

switch statement flow-diagram in C++

Here is the flow-diagram of a switch-statement,



Here is the program

Example of switch statement in C++ (with int)

```
#include<iostream>

using namespace std;

int main()
```

```
{

clrscr();

int input;

cout<<"Enter a number: ";

cin>>input;

switch(input)

{

case 1: cout<<"1st statement"; break;

case 2: cout<<"2nd statement"; break;

case 3: cout<<"3rd statement"; break;

default: cout<<"Try again..";

}

return 0;

}
```

OUTPUT

```
Enter a number: 2
```

2nd statement

Try with a different example here

Example of switch statement with char data-type

```
#include<iostream>

using namespace std;

int main()

{

    char input;

    cout<<"Enter a character: ";

    cin>>input;

    switch(input)

    {

        case 'a': cout<<"1st statement executed"; break;

        case 'b': cout<<"2nd statement executed";break;

        case 'c': cout<<"3rd statement executed";break;

        default: cout<<"Try again..";

    }

return 0;
```

```
}
```

OUTPUT

```
Enter a character: b
```

```
2nd statement executed
```

In the switch statement, an expression can have `int`, `char`, or special character types.

switch – statement में, एक expression value `int` , `char` या special character हो सकते हैं।

```
#include<iostream>

using namespace std;

int main()

{

    char input;

    cout<<"Enter any value: ";

    cin>>input;

    switch(input)

    {

        case 1: cout<<"1st statement"; break;

        case 'a': cout<<"entered value is a";break;
```

```
case 2: cout<<"entered value is 2";break;

case '/': cout<<" your entered / ";break;

case 'A': cout<<"entered value is A";break;

default: cout<<"Try again..";

}

return 0;

}
```

OUTPUT

```
Enter any value: A
```

```
entered value is A
```

Explanation

As we know that C++ is a sensitive language which means that it will treat “a” and “A” differently.

so even if the input value(a and A) here is the same, there will be a different output (due to different cases). but if we want same output with the same input values then codes will be as follows,

```
switch(char data-type)

{

case 'a':
```

```
    case 'A': cout<<"statement executed.."; break;

    .....

}
```

now case doesn't matter in the above code means this will give the same output even if a user entered "a" and "A".

creating a calculator using switch statement in C++

```
#include<iostream>

using namespace std;

int main()

{

    clrscr();

    int num1,num2;

    char input;

    cout<<"Enter any two number: ";

    cin>>num1>>num2;

    cout<<"Select operation(+ , - , * , /): ";

    cin>>input;

    switch(input)

    {
```

```
case '+': cout<<num1+num2; break;

case '-': cout<<num1-num2; break;

case '*': cout<<num1*num2; break;

case '/': cout<<num1/num2; break;

default: cout<<"Try again..";

}

return 0;

}
```

OUTPUT

```
Enter any two number: 2 6

Select operation(+ , - , * , /): +

8
```

NOTE: switch statement can not be of a float data type.

without break keyword switch statement

if there is no break keyword so,

```
switch(input)

{
```



```
case 1: cout<<"1st statement";

case 2: cout<<"2nd statement";

case 3: cout<<"3rd statement";

default: cout<<"Try again..";

}
```

now if the user-entered **number 1** then the OUTPUT will be as follows,

```
1st statement

2nd statement

3rd statement
```

means, all statements will be executed one by one. but if the user-entered **number 3** then OUTPUT will be different-

```
3rd statement
```

The reason for this is that execution in the switch-statement will first go to **case-1**, then **case-2** and then **case-3**. And as such, only in **case-3** is the condition becoming true. So this is the only single statement executed.

If there were more case-statements after **case-3**, they would all be executed (no matter if they were false or true). Since there is no statement after **case-3**, so only a single statement is being executed.

creating a menu using switch statement in C++

```
#include<iostream>

using namespace std;
```

```
int main()

{

int choice;

int num1,num2,result;

cout<<"1.Sum \n";

cout<<"2.Subtraction\n";

cout<<"3.Multiply\n";

cout<<"\nEnter Your Choice: ";

cin>>choice;

switch(choice)

{

case 1:

cout<<"\nEnter Two Number: ";

cin>>num1>>num2;

result = num1+num2;

cout<<"\nTotal: "<<result;

break;
```

```
case 2:

    cout<<"\nEnter Two Number: ";

    cin>>num1>>num2;

    result = num1-num2;

    cout<<"\nSubstraction: "<<result;

    break;

case 3:

    cout<<"\nEnter Two Number: ";

    cin>>num1>>num2;

    result = num1*num2;

    cout<<"\nMultiply: "<<result;

    break;

default:

    cout<<"\ninvalid choice....";

}
```

```
return 0;  
  
}
```

OUTPUT

```
1.Sum of two Number  
  
2.Subtraction of two Number  
  
3.Multiply of two Number  
  
Enter Your Choice:1  
  
Enter Two Number:4 5  
  
Total: 9
```

In a good programming practice, the same code should not be written repeatedly.

As we can see in the above program, the same statement has been repeated repeatedly in every case-statement,

```
cout<<"\nEnter Two Number: ";  
  
cin>>num1>>num2;
```

which would not be considered good programming practice.

This program is also created by the function in a better way in the Example of [function with switch in C++](#).

You have noticed, programs related to the switch behave like a menu. Therefore, be it C++, or any other language switch-statement is mostly used to create menus in the program.

