

C++ If Else

C++ If Else Statement

In C++, if else statements are used to perform conditional execution of statement(s). Based on the result of a condition, the decision to execute a block is taken.

In this tutorial, we shall learn the different forms of **if else** statement, their syntax with detailed explanation and examples for each of them.

In C++, If Else statement can occur in three forms. They are:

- If statement
- If-Else statement
- If-Else-If statement

C++ If Statement

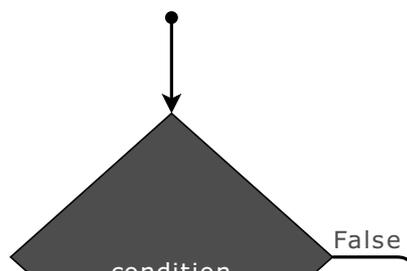
Following is the syntax of simple C++ If statement.

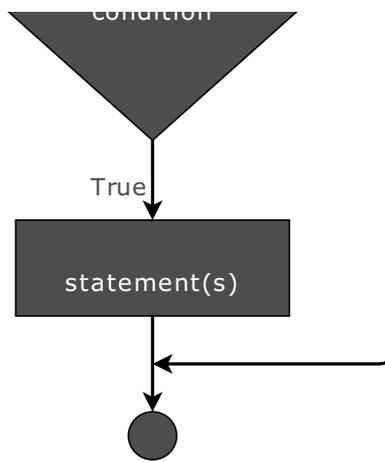
```
if (condition) {  
    // statement(s)  
}
```

If the condition is true, the statement(s) inside **if** block are executed. If the condition is false, the statement(s) inside **if** block are not executed. In either of the cases, the execution continues with the subsequent statements after the completion of **if** statement.

Flow Diagram

Following is the flow diagram of **if** statement in C++.





Example

Following is an example C++ program, where we use **if** statement to print a message only if the number is positive.

C++ Program

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

int main() {
    int a = 10;
    if (a>0) {
        cout << a << " is positive.";
    }
}
```

Output

```
10 is positive.
```

C++ If Else Statement

Following is the syntax of C++ If Else statement.

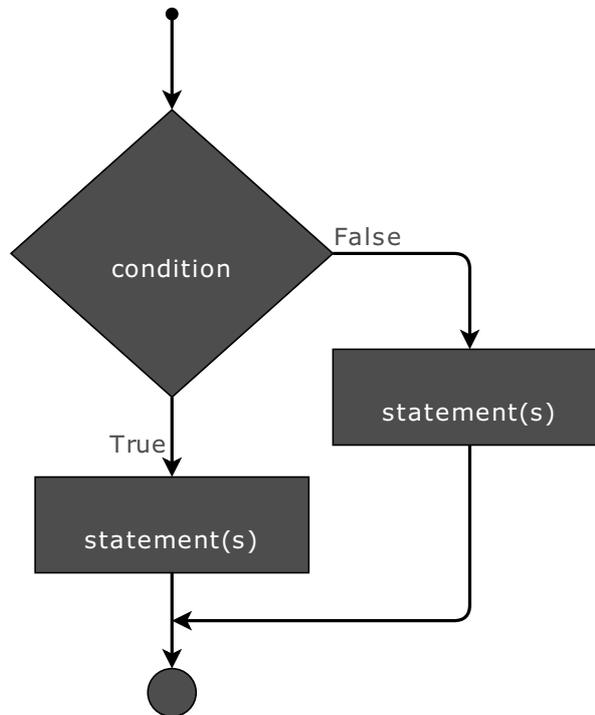
```
if (condition) {
    // statement(s)
} else {
    // statement(s)
}
```

If the condition is true, the **statement(s)** inside **if** block are executed. If the condition is false, the **statement(s)** inside **else** block are executed. After the execution of either **if** block or **else** block, program execution continues with next statements after **if-else** statement.

Else block is optional. So, if you do not provide **else** block to an **if** block, it becomes a simple **if** statement. So, **if-else** could be considered as an extension to simple **if** statement.

Flow Diagram

Following is the flow diagram of **if-else** statement in C++.



Example

In the following example C++ program, we use **if-else** statement to check if a number is positive or not.

C++ Program

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

int main() {
    int a = -10;
    if (a>0) {
        cout << a << " is positive.";
    } else {
        cout << a << " is not positive.";
    }
}
```

Output

```
-10 is not positive.
```

C++ If Else If Statement

Following is the syntax of C++ if-else-if statement.

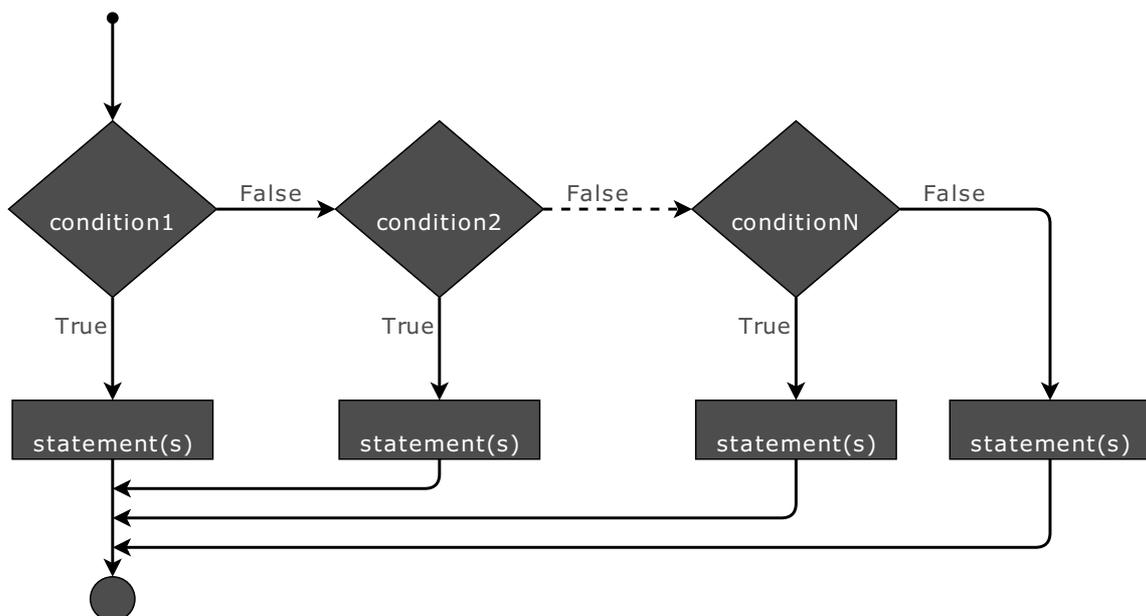
```
if (condition1) {  
    // statement(s)  
} else if (condition2) {  
    // statement(s)  
} else if (condition3) {  
    // statement(s)  
} else {  
    // statement(s)  
}
```

If condition1 is true, executes the statement(s) inside **if** block are executed. Else, the next condition in the else-if ladder is evaluated. The next condition is condition2. If the condition2 is true, the statement(s) inside corresponding **else if** block are executed. If condition2 is also false, condition3 is executed, and so on, until a condition is true. If no condition evaluates to true, then else block is executed.

Even in if-else-if statement, the last else block is optional.

Flow Diagram

Following is the flow diagram of **if-else-if** statement in C++.



Example

In the following example C++ program, we use **if-else-if** statement to check if a number is positive, negative or zero.

C++ Program

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

int main() {
    int a = -7;
    if (a>0) {
        cout << a << " is positive.";
    } else if(a<0) {
        cout << a << " is negaitive.";
    } else {
        cout << a << " is zero.";
    }
}
```

Output

```
-7 is negaitive.
```

Conclusion

In this [C++ Tutorial](#), we learned about different type of C++ If Else statement: simple If statement, if-else statement, if-else-if statement, with syntax, flow diagram and example for each one of them.

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