

# **Chapter Two**

## **Statements and flow control**

### **Part I**

- ❑ A simple C++ statement is each of the individual instructions of a program, like the variable declarations and expressions seen in previous sections.
- ❑ They always end with a semicolon (;), and are executed in the same order in which they appear in a program.
- ❑ But programs are not limited to a linear sequence of statements.
- ❑ During its process, a program may repeat segments of code, or take decisions and bifurcate.
- ❑ For that purpose, C++ provides flow control statements that serve to specify what has to be done by our program, when, and under which circumstances.

# C++ if Statement

- ❑ The **if** is a flow control statement used to make a decision based on a condition.
- ❑ In c++, the if statement is used to make decisions based on a condition.
- ❑ The if statement verifies the given condition and decides whether a block of statements to be executed or not based on the condition result.
- ❑ The if statement has the following forms.
  - if statement
  - if-else Statement
  - if-else if Statement

# **C++ if Statement**

## if Statement

The if statement checks the given condition and executes the block of code if the condition is evaluated to true. In other words, the if statement is used to execute a block of code if the given condition is true.

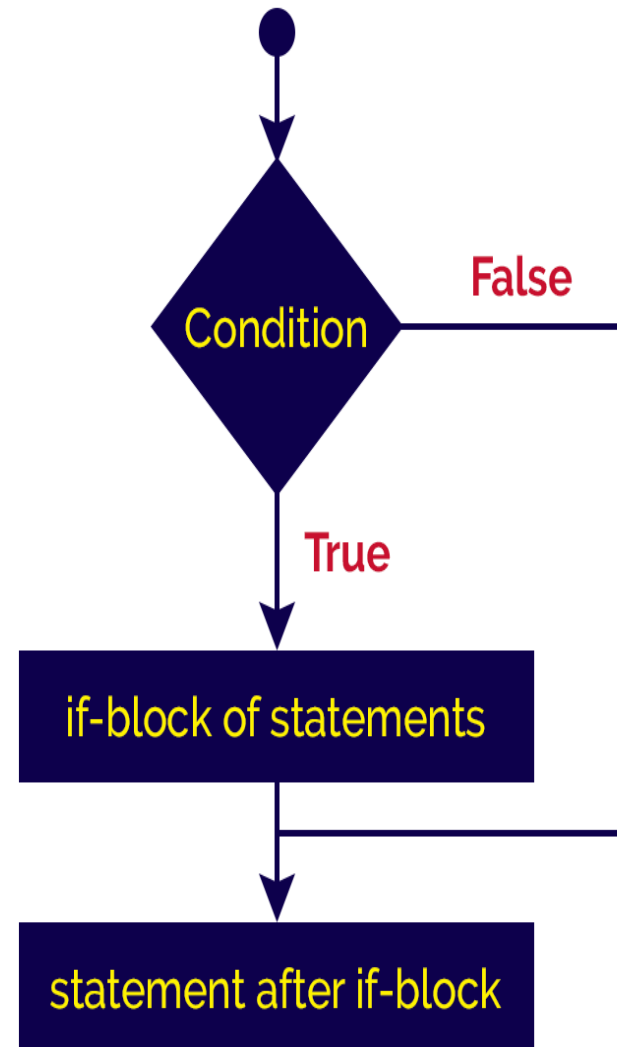
## Syntax - if Statement

```
if (condition) { // block of code to be executed if the  
condition is true }
```

## Syntax

```
if(condition){  
    if-block of statements;  
    ...  
}  
statement after if-block;  
...
```

## Flow of execution



## C++ if Statement...

- ❑ In any “if” statement, first the “expression” will be evaluated and if the outcome is non zero (which means TRUE), then the “statements” is executed. Otherwise, nothing happens (the statement will not be executed) and execution resumes to the line immediately after the “if” block.
- ❑ To make multiple statements dependent on the same condition we can use a compound statement, which will be implemented by embracing the group of statements with left “{” and right “}” French

E.g.:

```
if(age>18)
    cout<<"you are an adult";
```

E.g.:

```
if(balance > 0)
{
    interest = balance * creditRate;
    balance += interest;
}
```

## **C++ if Statement...**

- ❑ Most of the time “expression” will have relational expressions testing whether something is equal, greater, less, or different from something else. □
- ❑ It should be noted that the output of a relational expression is either True (represented by anything different from zero) or False (represented by Zero). □
- ❑ Thus any expression, whose final result is either zero or none zero can be used in “expression”

## C++ if Statement...

- ❑ Most of the time “expression” will have relational expressions testing whether something is equal, greater, less, or different from something else. □
- ❑ It should be noted that the output of a relational expression is either True (represented by anything different from zero) or False (represented by Zero). □
- ❑ Thus any expression, whose final result is either zero or non zero can be used in “expression”

E.g.:

```
int x;  
cin>>x;  
if(x)  
    cout<<"you are an adult";
```

## C++ if Statement...

- ❑ In the above example, the value of variable x will be an input from the user. The “if” statement will be true if the user gives anything different from zero, which means the message “you are an adult” will be displayed on the screen. If the user gives zero value for x, which will be interpreted as False, nothing will be done as the if statement is not satisfied. ❑
- ❑ Thus, expression can be:
  - Relational expression,
  - A variable,
  - A literal constant, or
  - Assignment operation, as the final result is whatever we have at the right hand side and the one at the left hand side is a variable.

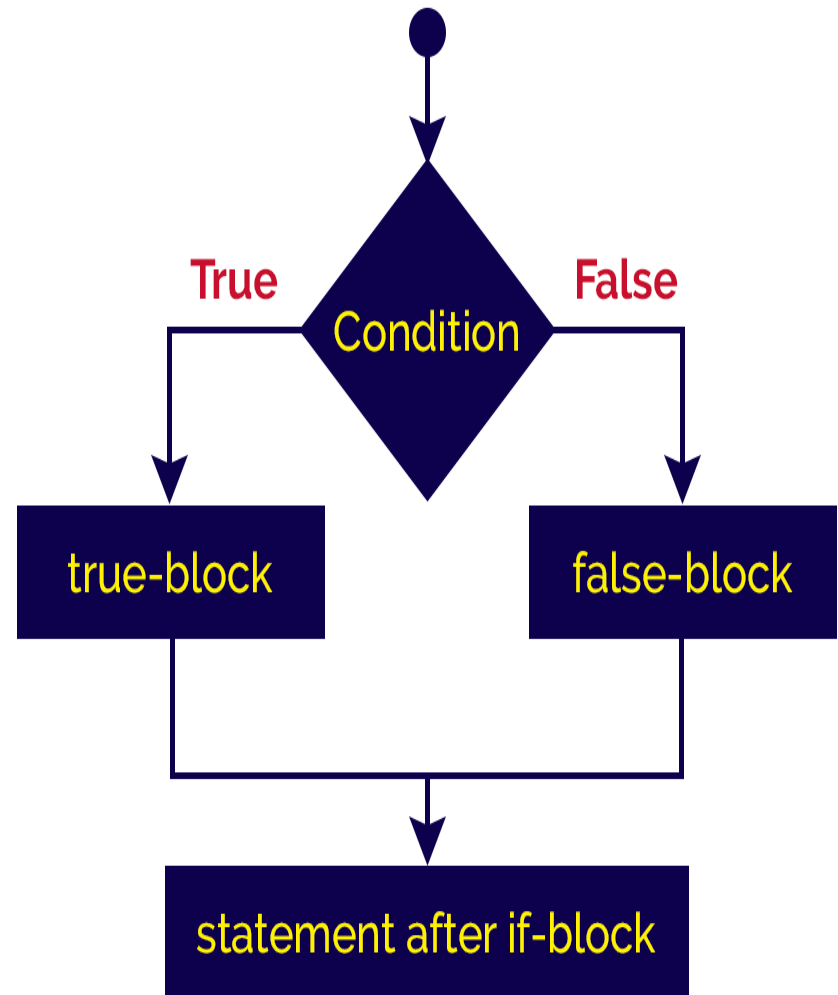
## **The If else statement**

- ❑ Another form of the “if” is the “if ...else” statement.
  - ❑
- ❑ The “if else if” statement allows us to specify two alternative statements:
  - One which will be executed if a condition is satisfied and
  - Another which will be executed if the condition is not satisfied.

# Syntax

```
if(condition){  
    true-block of statements;  
    ...  
}  
else{  
    false-block of statements;  
    ...  
}  
statement after if-block;  
...
```

# Flow of execution



## Example - Code to find largest of two numbers

```
#include <iostream>

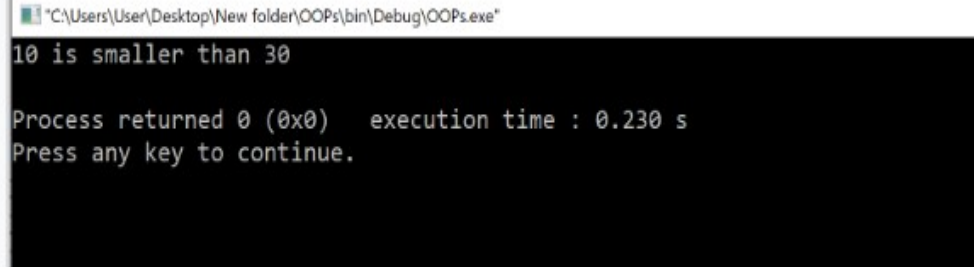
using namespace std;

int main()
{
    int a = 10, b = 30;

    if(a < b) {
        cout << a << " is smaller than " << b << endl;
    } else {
        cout << a << " is larger than " << b << endl;
    }

    return 0;
}
```

### Output



```
"C:\Users\User\Desktop\New folder\OOPs\bin\Debug\OOPs.exe"
10 is smaller than 30

Process returned 0 (0x0)   execution time : 0.230 s
Press any key to continue.
```

# The “if else if” statement

- ❑ The if-elseif statement used to check a new condition in the elseif if the first condition is evaluated to false.
- ❑ The elseif statement must follow the if statement

## Syntax - if-elseif Statement

```
if (condition_1) {  
    // block of code to be executed if the condition_1 is true  
} else if (condition_2) {  
    // block of code to be executed if the condition_2 is true  
}
```

- ❑ The condition\_2 is verified only when condition\_1 is false.

## Example - Code to find largest of three numbers

```
#include <iostream>

using namespace std;

int main()
{
    int a = 10, b = 30, c = 20;

    if(a > b && a > c) {
        cout << a << " is larger than " << b << " and " << c << endl;
    } else if( b > c) {
        cout << b << " is larger than " << a << " and " << c << endl; // execution
    } else {
        cout << c << " is larger than " << a << " and " << b << endl;
    }

    return 0;
}
```

"C:\Users\User\Desktop\New folder\OOPs\bin\Debug\OOPs.exe"

30 is larger than 10 and 20

Process returned 0 (0x0) execution time : 0.297 s

Press any key to continue.

# Nested if statements

It is always legal to **nest** if-else statements, which means you can use one if or else if statement inside another if or else if statement(s).

## Syntax

The syntax for a **nested if** statement is as follows –

```
if( boolean_expression 1) {  
    // Executes when the boolean expression 1 is true  
    if(boolean_expression 2) {  
        // Executes when the boolean expression 2 is true  
    }  
}
```

You can nest **else if...else** in the similar way as you have nested *if* statement.

## Example

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

int main () {
    // local variable declaration:
    int a = 100;
    int b = 200;

    // check the boolean condition
    if( a == 100 ) {
        // if condition is true then check the following
        if( b == 200 ) {
            // if condition is true then print the following
            cout << "Value of a is 100 and b is 200" << endl;
        }
    }
    cout << "Exact value of a is : " << a << endl;
    cout << "Exact value of b is : " << b << endl;

    return 0;
}
```

Value of a is 100 and b is 200

Exact value of a is : 100

Exact value of b is : 200

**Cont**

