

1-Getting start with statement in C++

- Program heading
- Begin type or variable declartion
- Statement of operation
- Results

Example:

```
# include<iostream>.....librires
```

```
Main ().....function
```

```
{
```

```
Step1;
```

```
Step2;
```

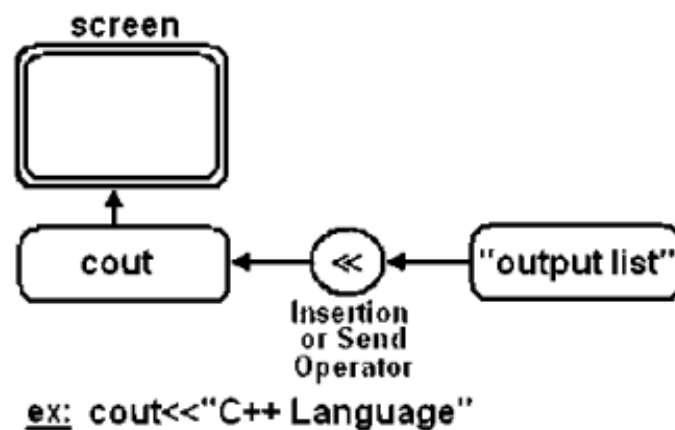
```
Step n ;
```

```
}
```

The keyboard and screen I/O instruction in C++ are:

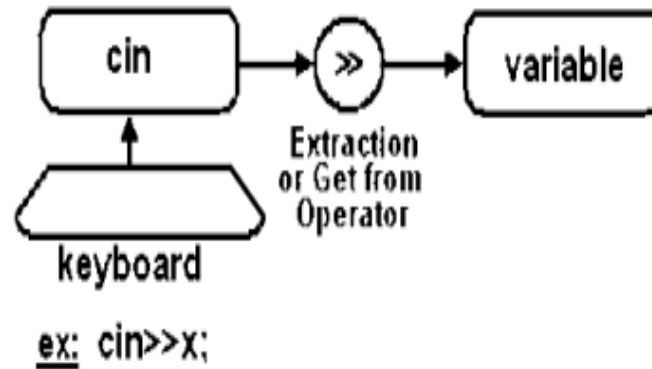
1- Display an object on

```
Cout <<var1<<var2<<var3
```



2-Cin:is used to read an object from a standard input device(keyboard)

Cin>>var1>>var2>>var3



<< The stream instertion operator(or send operator)

>> The stream extraction operator (or get from operator).

; semicolon, the terminator of every C++ statement.

The **endl** is used in c++ to represent a new line, as shown in the following example:

Example 2

```
#include<iostream.h>
void main( )
{
    cout << "hallow" << endl;
    cout << "students";
}
```

Output:

hallow
students

The `\n` is a special escape code, also used in C++ to represent a new line, as shown in the following example:

Example 3	
<pre>#include<iostream.h> void main() { cout << "hallow \n"; cout << "students"; }</pre>	<p>Output:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>hallow students</pre> </div>

2- Statement

A statement in a computer carries out some action, there are three types of statement used in c++, they are expression statement, compound statement and control statement

Expression statement	Compound statement	Control statement
<pre>x=y; sum=x+y;</pre>	<pre>{ a=b+c; x=x*x; y=a+x; }</pre>	<pre>if (a>b) { a=l; k=a+1; }</pre>

Conditional Statements

Conditional statements let you execute code based on whether or not something is true. As shown in the following sections, there are three main types of conditional statements in C++:

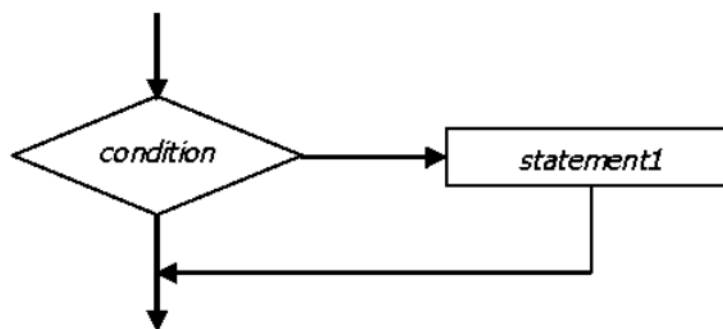
1. If/else statements.
2. Switch statements.
3. Conditional Statement

1-If/else Statements:

The IF statement is used to express conditional expression. If the given condition is true then it will execute the statements; otherwise it will execute the optional statements.

General Form of single-selection If statement:

```
if ( expression or condition ) statement1 ;
```



There are three forms of if...else statements in C++.

- if statement
- if...else statement
- if...else if...else statement

The most common conditional statement is the if statement, which can be accompanied by an else. If the condition given inside the if statement is true, the line or block of code is executed. If not, execution continues with the else case if present, or with the code following the conditional statement. The following code shows a cascading if statement.



In computer programming, we use the if...else statement to run one block of code under certain conditions and another block of code under different conditions.

```
If ( I > 4 ) {  
    // Do something.  
} else if ( I > 2 ) {  
    // Do something else.  
} else {  
    // Do something else.  
}
```

For example, assigning grades (A, B, C) based on marks obtained by a student.

if the percentage is above 90, assign grade A

if the percentage is above 75, assign grade B

if the percentage is above 65, assign grade C

The if statement evaluates the condition inside the parentheses ().

- If the condition evaluates to **true**, the code inside the body of if is **executed**.
- If the condition evaluates to **false**, the code inside the body of if is **skipped**.

Logical Evaluation Operators

You have already seen a logical evaluation operator without a formal definition.

The > operator compares two values. The result is “true” if the value on the left is greater than the value on the right. All logical evaluation operators follow this

pattern, they all result in a true or false. The following table shows common logical evaluation operators:

OP	DESCRIPTION	USAGE
< <= > >=	Determines if the left-hand side is less than, less than or equal to, greater than, or greater than or equal to the right-hand side	if (i < 0) { std::cout << "i is negative"; }
==	Determines if the left-hand side equals the right-hand side. Don't confuse this with the = (assignment) operator!	if (i == 3) { std::cout << "i is 3"; }
!=	Not equals. The result of the statement is true if the left-hand side does <i>not</i> equal the right-hand side.	if (i != 3) { std::cout << "i is not 3"; }

OP	DESCRIPTION	USAGE
!	Logical NOT. This complements the true/false status of a Boolean expression. This is a unary operator.	if (!someBoolean) { std::cout << "someBoolean is false"; }
&&	Logical AND. The result is true if both parts of the expression are true.	if (someBoolean && someOtherBoolean) { std::cout << "both are true"; }
	Logical OR. The result is true if either part of the expression is true.	if (someBoolean someOtherBoolean) { std::cout << "at least one is true"; }

Example :

// Program to print positive number entered by the user

// If the user enters a negative number, it is skipped

```
#include <iostream>
```

```
int main() {
```

```
    int number;
```

```
    cout << "Enter an integer: ";
```

```
    cin >> number;
```



// checks if the number is positive

```
if (number > 0) {  
    cout << "You entered a positive integer: " << number << endl;  
}  
cout << "This statement is always executed.";  
return 0;  
}
```

The output :

```
Enter an integer: 7  
You entered a positive integer: 7  
This statement is always executed.
```

When the user enters 7, the condition $number > 0$ is evaluated to true, and the statement inside the body of if is executed.

```
Enter an integer: -1  
This statement is always executed.
```

When the user enters -1, the condition $number > 0$ is evaluated to false and the statement inside the body of if is not executed.

2-C++ if...else

The if statement can have an optional else clause. Its syntax is:

```
if (condition) {  
    // block of code if condition is true  
}  
else {  
    // block of code if condition is false  
}
```

If the condition evaluates true,

- the code inside the body of if is executed
- the code inside the body of else is skipped from execution

If the condition evaluates false,

- the code inside the body of else is executed
- the code inside the body of if is skipped from execution

Example 2: C++ if...else Statement

// Program to check whether an integer is positive or negative

// This program considers 0 as a positive number

```
#include <iostream>  
using namespace std;  
int main() {
```




```
int number;

cout << "Enter an integer: ";
cin >> number;

if (number >= 0) {
    cout << "You entered a positive integer: " << number << endl;
}
else {
    cout << "You entered a negative integer: " << number << endl;
}
cout << "This line is always printed.";
return 0;
}
```

Output 1

```
Enter an integer: 4
You entered a positive integer: 4.
This line is always printed.
```

In the above program, we have the condition $number \geq 0$. If we enter the number greater or equal to 0, then the condition evaluates true.

Here, we enter 4. So, the condition is true. Hence, the statement inside the body of if is executed.


Output 2

```
Enter an integer: -4
You entered a negative integer: -4.
This line is always printed.
```

Here, we enter -4. So, the condition is false. Hence, the statement inside the body of else is executed.


Example :

Example 3

 Write a C++ program to read a student degree, and check if it's degree greater than or equal to 50, then print pass, otherwise print fail:

```
#include<iostream.h>
void main( )
{
    int degree;
    cin >> degree;
    if (degree >= 50 )
        cout << "pass";
    else
        cout << "fail";
}
```

Example 5

 Write a C++ program to read a number, and print the day of the week:

```
#include<iostream.h>
void main( )
{
    int day;
    cin >> day;
    if ( day == 1 ) cout << "Sunday";
    else if (day == 2 ) cout << "Monday";
    else if (day == 3 ) cout << "Tuesday";
    else if (day == 4 ) cout << "Wednesday";
    else if (day == 5 ) cout << "Thursday";
    else if (day == 6 ) cout << "Friday";
    else if (day == 7 ) cout << "Saturday";
    else cout << "Invalid day number";
}
```