Lab # 3

Relational and logical operator

There come situations in real life when we need to make some decisions and based on these decisions, we decide what we should do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code.

Decision making statements in programming languages decides the direction of flow of program execution. Decision making statements available in C++ are:

* [if statement](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/#if)
* [if..else statements](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/#if-else)
* [nested if statements](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/#nested-if)
* [switch statements](https://www.geeksforgeeks.org/switch-statement-cc/)

if statement

if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

Syntax:

if(condition)

{

// Statements to execute if

// condition is true

}

Here, **condition**after evaluation will be either true or false. if statement accepts boolean values – if the value is true then it will execute the block of statements below it otherwise not.  
If we do not provide the curly braces ‘{‘ and ‘}’ after if( condition ) then by default if statement will consider the first immediately below statement to be inside its block.  
**Example**:

if(condition)

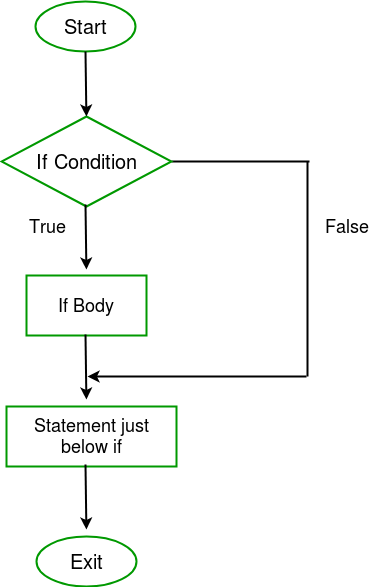
statement1;

statement2;

// Here if the condition is true, if block

// will consider only statement1 to be inside

// its block.

**Flowchart**:  


|  |
| --- |
| // C++ program to illustrate If statement  #include<iostream>  using namespace std;        int main()      {          int i = 10;            if (i > 15)          {             cout<<"10 is less than 15";          }            cout<<"I am Not in if";      } |
| // C++ program to illustrate If statement      } |

Output:

I am Not in if

As the condition present in the if statement is false. So, the block below the if statement is not executed.

if- else

The *if*statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won’t. But what if we want to do something else if the condition is false. Here comes the *else*statement. We can use the *else*statement with *if*statement to execute a block of code when the condition is false.  
**Syntax**:

if (condition)

{

// Executes this block if

// condition is true

}

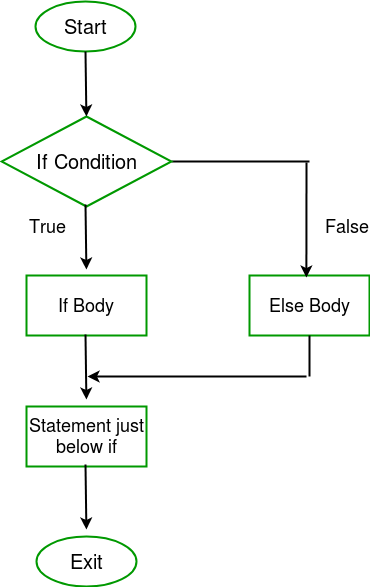
else

{

// Executes this block if

// condition is false

}

**Flowchart**:  
  
**Example**:

|  |
| --- |
| // C++ program to illustrate if-else statement  #include<iostream>  using namespace std;    int main()   {          int i = 20;            if (i < 15)              cout<<"i is smaller than 15";          else              cout<<"i is greater than 15";        return 0;   } |

Output:

i is greater than 15

The block of code following the *else*statement is executed as the condition present in the *if*statement is false.

Nested-if

A nested if is an if statement that is the target of another if statement. Nested if statements means an if statement inside another if statement. Yes, C++ allows us to nest if statements within if statements. i.e, we can place an if statement inside another if statement.  
**Syntax**:

if (condition1)

{

// Executes when condition1 is true

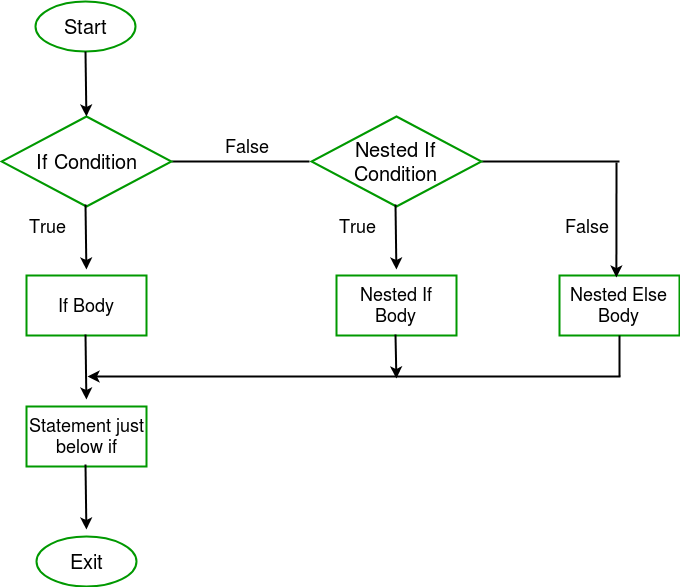
if (condition2)

{

// Executes when condition2 is true

}

}

**Flowchart**:  
  
**Example**:

|  |
| --- |
| // C++ program to illustrate nested-if statement  int main()      {          int i = 10;            if (i == 10)          {              // First if statement              if (i < 15)                 cout<<"i is smaller than 15";                // Nested - if statement              // Will only be executed if statement above              // it is true              if (i < 12)                  cout<<"i is smaller than 12 too";              else                  cout<<"i is greater than 15";          }            return 0;      } |

Output:

i is smaller than 15

i is smaller than 12 too

Tasks:

1. Write a program to check whether a number is positive, negative or zero?
2. Write a program to check whether a character is uppercase or lower case?
3. [Write a program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:](https://codeforwin.org/2015/05/c-program-to-enter-student-marks-and-calculate-percentage-and-grade.html)  
   Percentage >= 90% : Grade A  
   Percentage >= 80% : Grade B  
   Percentage >= 70% : Grade C  
   Percentage >= 60% : Grade D  
   Percentage >= 40% : Grade E  
   Percentage < 40% : Grade F