Lab No.4

Conditions (if, if-else, nested if-else)

Objectives

- To understand Multiple Nested If-Else, If statement.
- Logical Operators and flowcharts
- Design and Compilation of C++ programs based on decision making.

In programming also where we need to make some decisions and based on these decisions we will execute the next block of code. For example, in C++ if x occurs then execute y else execute z. There can also be multiple conditions like if x occurs then execute p, else if condition y occurs execute q, else execute r.

if statement in C/C++

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:

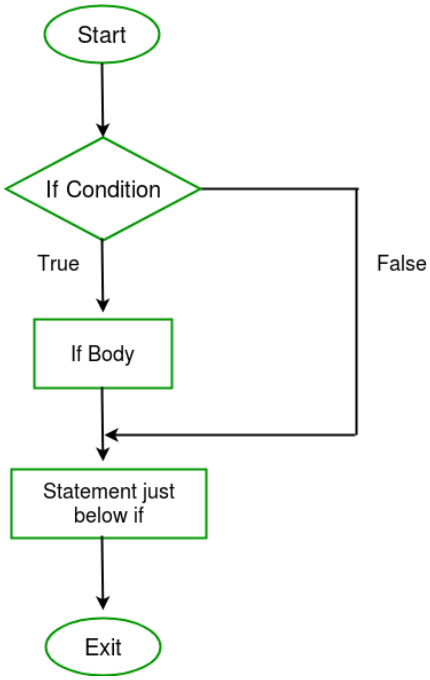
```
if(condition)
{
// Statements to execute if
// condition is true
}
```

Here, the 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.
```

Flow chart of if statement



Example-01:

```
// C program to illustrate If statement
#include <stdio.h>

int main() {
int i = 10;

if (i > 15)
{
printf("10 is less than 15");
}

printf("I am Not in if");
}
```

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

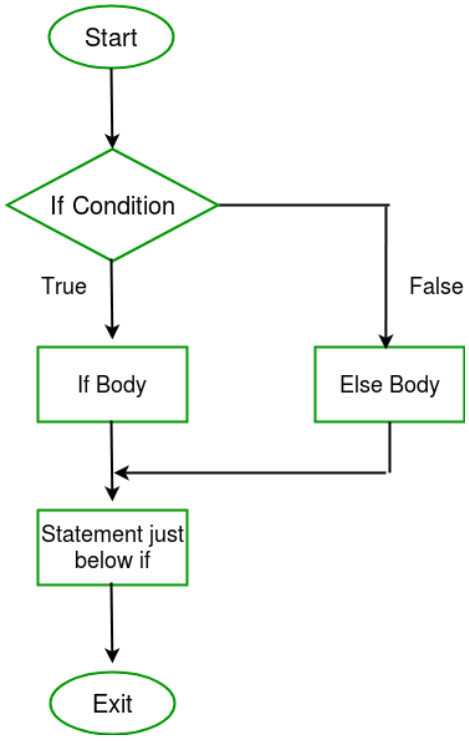
If-else in C/C++

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
}
```

Flow chart of if-else statement



Example-02:

```
// C program to illustrate If statement
#include <stdio.h>

int main() {
int i = 20;

if (i < 15){

printf("i is smaller than 15");
}
else{
```

```
printf("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.

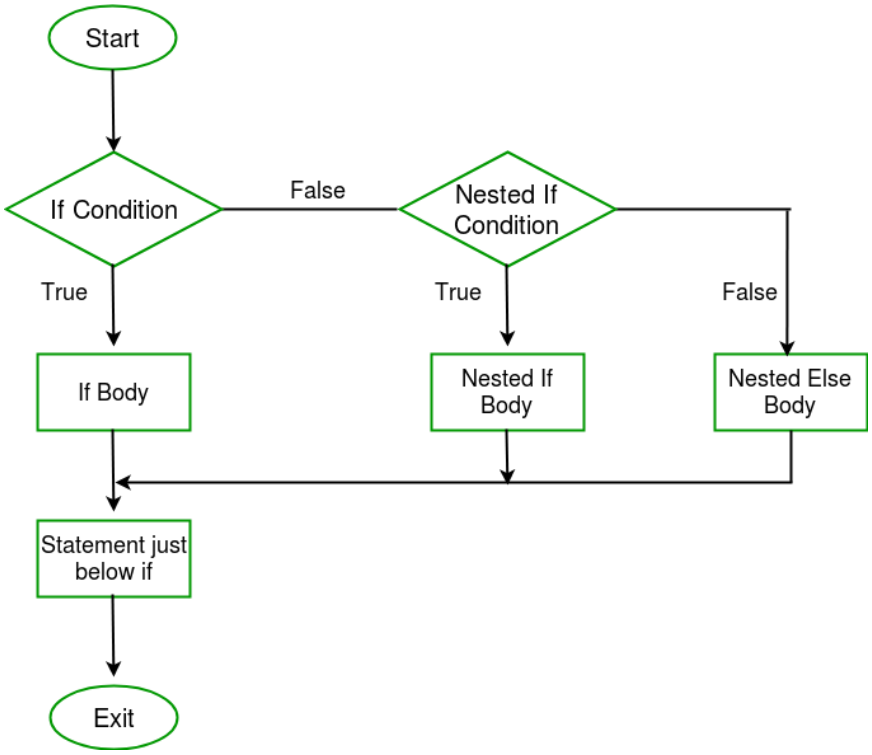
nested-if in C/C++

A nested-*if* in C++ is an *if* statement that is the target of another *if* statement. Nested *if* statements mean an *if* statement inside another *if* statement.

Syntax:

```
if (condition1)
{
// Executes when condition1 is true
if (condition2)
{
// Executes when condition2 is true
}
}
```

Flow chart of nested-if statement



Example-03:

```
// C program to illustrate nested-if statement
#include <stdio.h>

int main() {
int i = 10;

if (i == 10)
{
// First if statement
if (i < 15)
printf("i is smaller than 15\n");

// Nested - if statement
// Will only be executed if statement above
// is true
if (i < 12)
printf("i is smaller than 12 too\n");
else
printf("i is greater than 15");
}

return 0;
}
```

Output:

```
i is smaller than 15

i is smaller than 12 too
```

if-else-if ladder in C/C++

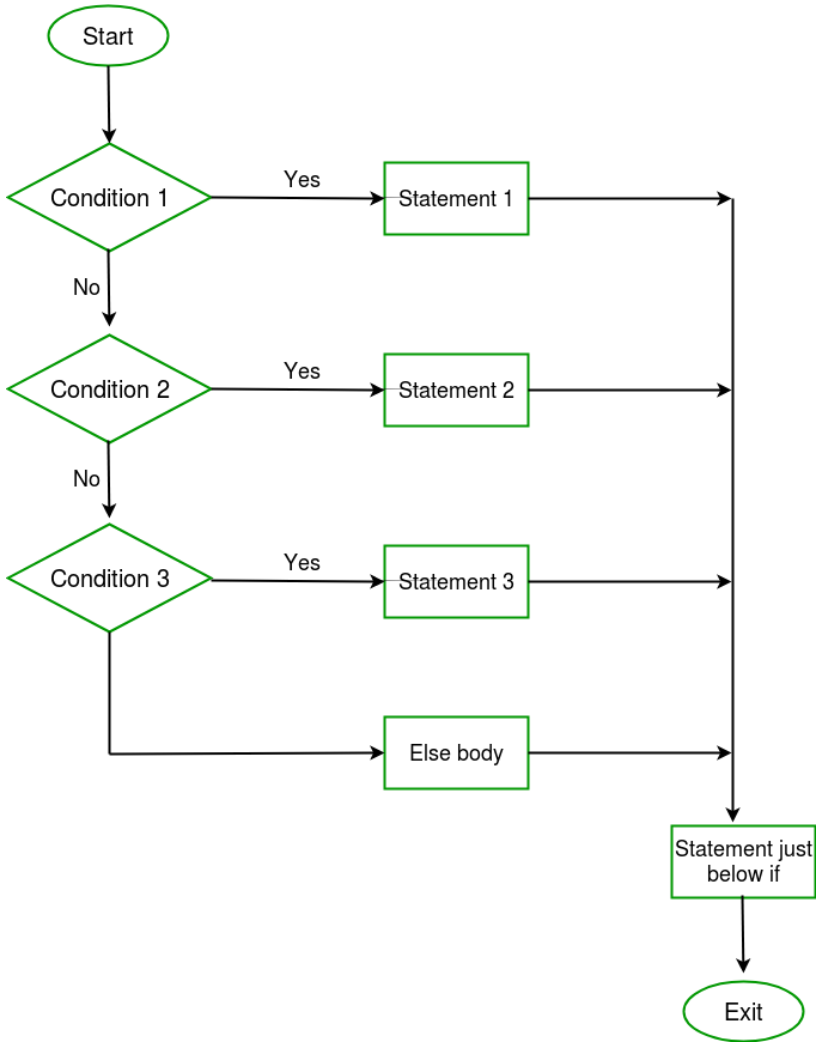
Here, a user can decide among multiple options. *if* statements are executed from the top down. As soon as one of the conditions controlling the *if* is true, the statement associated with that *if* is executed, and the rest of the *else-if* ladder is bypassed. If none of the conditions are true, then the final else statement will be executed.

Syntax:

```
if (condition)
    statement;
else if (condition)
```

```
    statement;
.
.
else
    statement;
```

Flow chart of if-else-if statement



Example-04

```
// C program to illustrate nested-if statement
#include <stdio.h>

int main() {
    int i = 20;

    if (i == 10)
        printf("i is 10");
```

```
    else if (i == 15)
        printf("i is 15");
    else if (i == 20)
        printf("i is 20");
    else
        printf("i is not present");
}
```

Output:

```
i is 20
```

LAB TASKS

Task-01:

Write a program in C++ that take input of three integer's numbers from user. Find the largest number among three of them.

Task-02:

Grade Program using nested if else, Write a program in C++ using if/else operator with nested statements to find the grade of a student. The detail is as follow.

grade>=90

- **Grade A**

grade>=80

- **Grade B**

grade>=70

- **Grade C**

grade>=60

- **Grade D**

grade< 60

- **Grade F.**

Task-03:

There are 5 vowels, that are **a, e, i, o, u** or **A, E, I, O, U**. On the basis of these 5 vowels, create the program, that checks whether input character by user at run-time is a vowel or not.

Task-04:

Write a program to check whether a entered character is lowercase (a to z) or uppercase (A to Z).

Task-05:

A student will not be allowed to sit in the exam if his/her attendance is less than 75%. The student will be only allowed to sit if he/she has a medical cause. Ask the user if he/she has a medical cause (Y or N). Take the following input from the user:

a. No. of classes held
b. No. of classes attended
c. Print % of classes attended and if the student is allowed to sit in the exam or not. Print accordingly.
