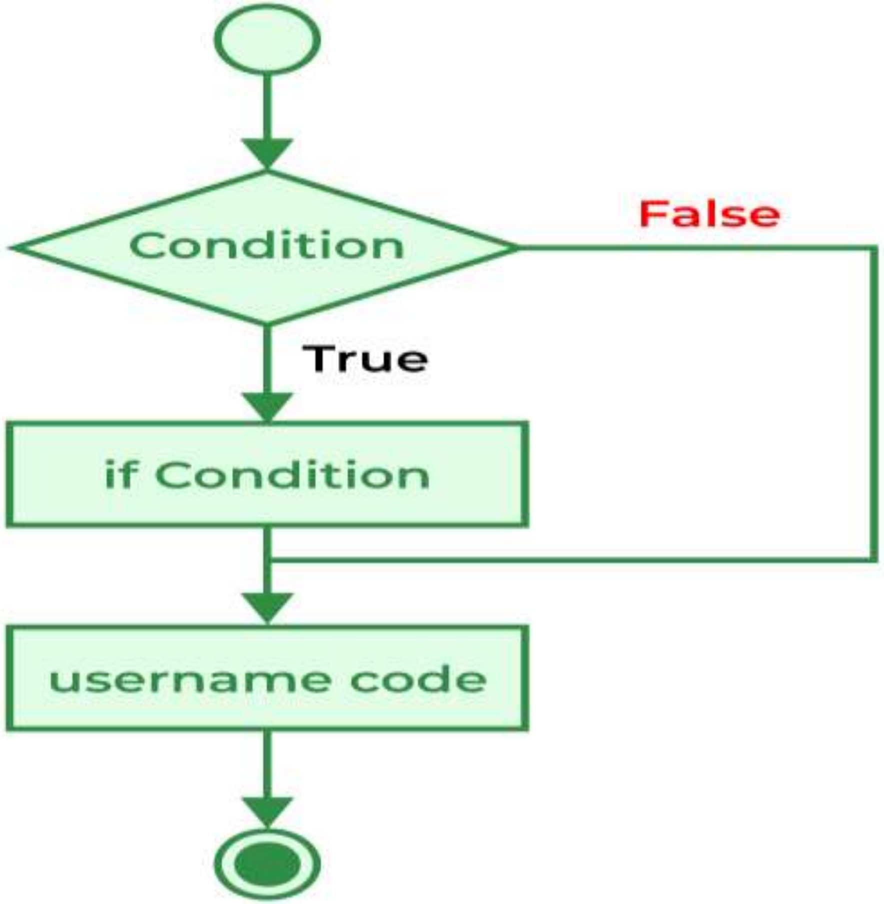
- 1. if in C/C++
- The if statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statements is executed otherwise not.
- Syntax of if Statement
- if(condition)
- {
-    // Statements to execute if
-    // condition is true
- }
- Here, the condition after evaluation will be either true or false. C 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.


Condition
False
True
if Condition
username code

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

int main()
{
    int i = 10;

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

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

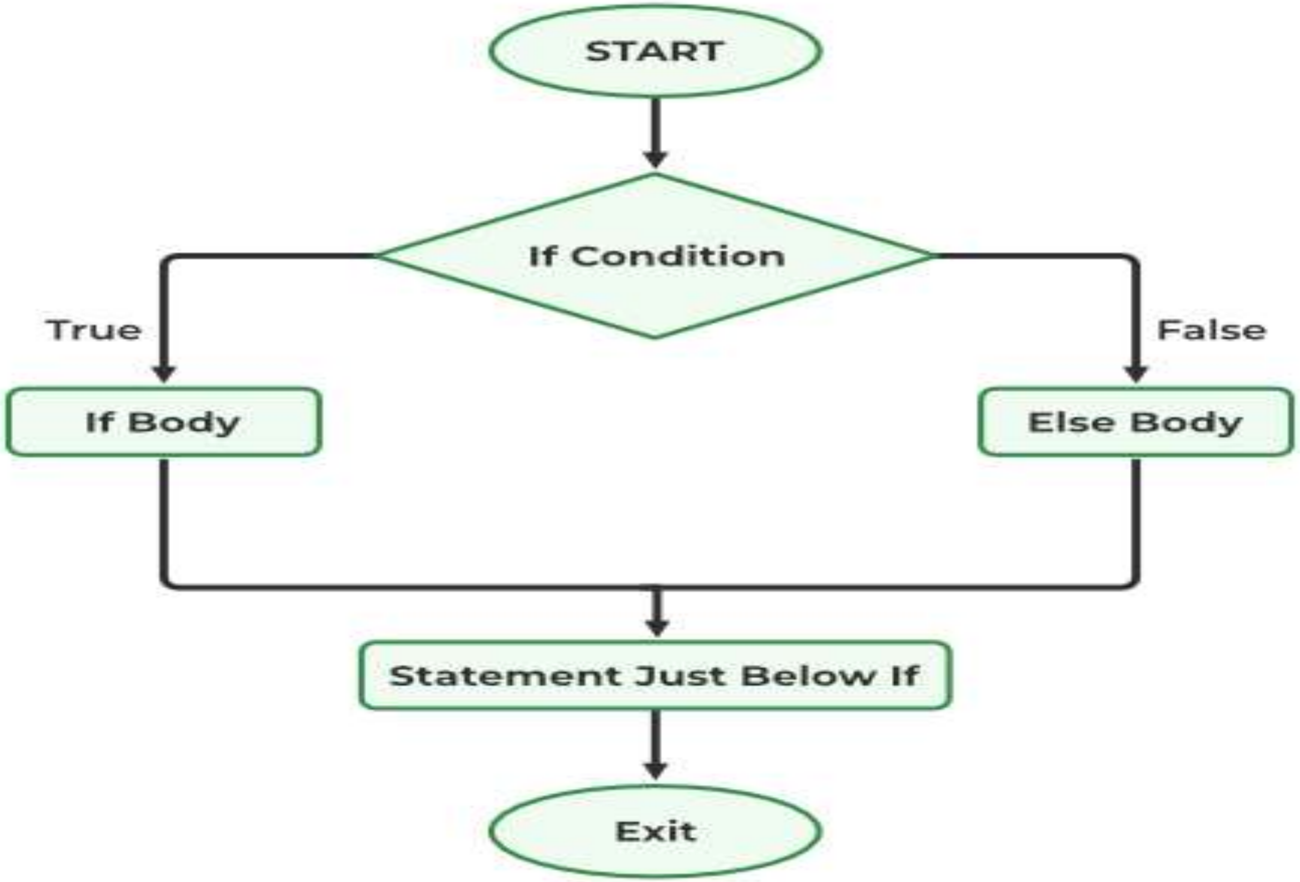
Output

- 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.

2. 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 when the condition is false? Here comes the C else statement.
- We can use the else statement with the if statement to execute a block of code when the condition is false. The if-else statement consists of two blocks, one for false expression and one for true expression.

- Syntax of if else in C/C++
- if (condition)
- {
- // Executes this block if
- // condition is true
- }
- else
- {
- // Executes this block if
- // condition is false
- }


START
If Condition
True
False
If Body
Else Body
Statement Just Below If
Exit

```
// 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.

if-else-if Ladder in C/C++

- The if else if statements are used when the user has to decide among multiple options. The C 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 C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. if-else-if ladder is similar to the switch statement.
- Syntax of if-else-if Ladder
- if (condition)
-     statement;
- else if (condition)
-     statement;
- else
-     statement;

The picture can't be displayed.

```
• // 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");
• }
```