

# If, If-Else, Nested If-Else and Ladder Else-If in C

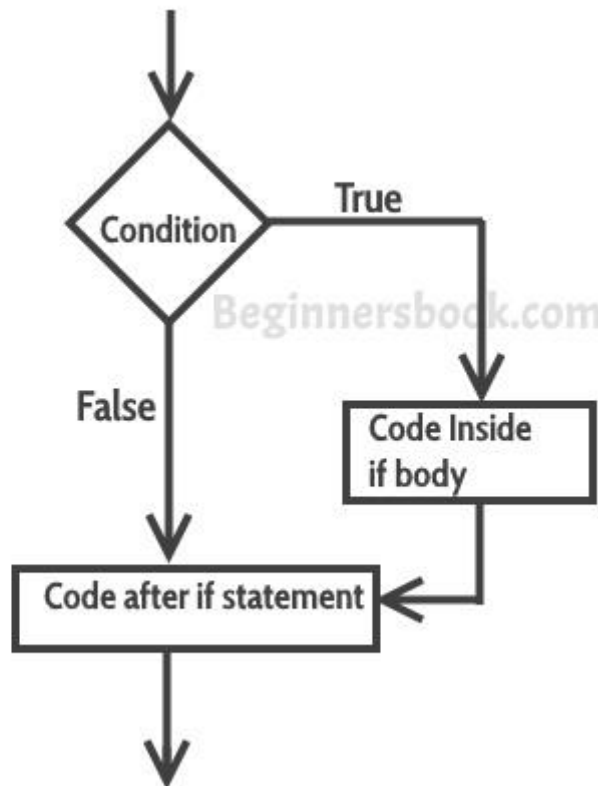
## 1) If (only) statement

Syntax of if statement:

The statements inside the body of “if” only execute if the given condition returns true. If the condition returns false then the statements inside “if” are skipped.

```
if (condition)
{
    //Block of C statements here
    //These statements will only execute if the condition is true
}
```

### Flow Diagram of if statement



## Example of if statement

```
#include <stdio.h>
int main()
{
    int x = 20;
    int y = 22;
    if (x<y)
    {
        printf("Variable x is less than y");
    }
    return 0;
}
```

### Output:

```
Variable x is less than y
```

**Explanation:** The condition (x<y) specified in the “if” returns true for the value of x and y, so the statement inside the body of if is executed.

## Example of multiple if statements

We can use multiple if statements to check more than one conditions.

```
#include <stdio.h>
int main()
{
    int x, y;
    printf("enter the value of x:");
    scanf("%d", &x);
    printf("enter the value of y:");
    scanf("%d", &y);
    if (x>y)
    {
        printf("x is greater than y\n");
    }
    if (x<y)
    {
        printf("x is less than y\n");
    }
    if (x==y)
    {
        printf("x is equal to y\n");
    }
    printf("End of Program");
    return 0;
}
```

In the above example the output depends on the user input.

Output: For example the user entered the same value for both x and y;

```
enter the value of x:20
enter the value of y:20

x is equal to y

End of Program
```

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

## C - If..else, Nested If-else and else-If (Ladder) Statement with example

In the last tutorial we learned how to use if statement in C. In this guide, we will learn how to use if else, nested if else and else if statements in a C Program.

### 2) *If-else* statement

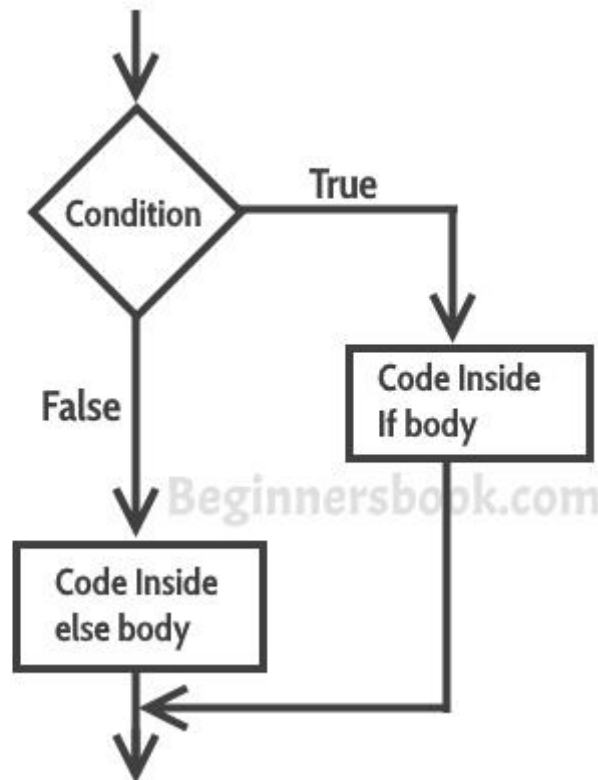
#### **Syntax of if else statement:**

If condition returns true then the statements inside the body of “if” are executed and the statements inside body of “else” are skipped.

If condition returns false then the statements inside the body of “if” are skipped and the statements in “else” are executed.

```
if(condition) {
    // Statements inside body of if
}
else {
    //Statements inside body of else
}
```

## Flow diagram of if else statement



## Example of *if-else* statement

In this program user is asked to enter the age and based on the input, the if-else statement checks whether the entered age is greater than or equal to 18. If this condition meet then display message “You are eligible for voting”, however if the condition doesn’t meet then display a different message “You are not eligible for voting”.

```
#include <stdio.h>
int main()
{
    int age;
    printf("Enter your age:");
    scanf("%d",&age);
    if(age >=18)
    {
        /* This statement will only execute if the
        * above condition (age>=18) returns true
        */
        printf("You are eligible for voting");
    }
    else
```

```

{
    /* This statement will only execute if the
    * condition specified in the "if" returns false.
    */
    printf("You are not eligible for voting");
}
return 0;
}

```

Output:

```

Enter your age:14
You are not eligible for voting

```

**Note:** If there is **only one statement** is present in the “if” or “else” body then you do not need to use the braces (parenthesis). For example the above program can be rewritten like this:

```

#include <stdio.h>
int main()
{
    int age;
    printf("Enter your age:");
    scanf("%d",&age);
    if(age >=18)
        printf("You are eligible for voting");
    else
        printf("You are not eligible for voting");
    return 0;
}

```

### 3) Nested *If-else* statement

When an if-else statement is present inside the body of another “if” or “else” then this is called nested if else.

**Syntax of Nested if else statement:**

```

if(condition) {
    //Nested if else inside the body of "if"
    if(condition2) {
        //Statements inside the body of nested "if"
    }
    else {
        //Statements inside the body of nested "else"
    }
}

```

```

    }
}
else {
    //Statements inside the body of "else"
}

```

## Example of nested *if-else*

```

#include <stdio.h>
int main()
{
    int var1, var2;
    printf("Input the value of var1:");
    scanf("%d", &var1);
    printf("Input the value of var2:");
    scanf("%d",&var2);
    if (var1 != var2)
    {
        printf("var1 is not equal to var2\n");
        //Nested if else
        if (var1 > var2)
        {
            printf("var1 is greater than var2\n");
        }
        else
        {
            printf("var2 is greater than var1\n");
        }
    }
    else
    {
        printf("var1 is equal to var2\n");
    }
    return 0;
}

```

Output:

```

Input the value of var1:12
Input the value of var2:21
var1 is not equal to var2
var2 is greater than var1

```

## 4) *else-if (Ladder)* statement

The else..if statement is useful when you need to check multiple conditions within the program, nesting of if-else blocks can be avoided using else..if statement.

## Syntax of else..if statement:

```
if (condition1)
{
    //These statements would execute if the condition1 is true
}
else if(condition2)
{
    //These statements would execute if the condition2 is true
}
else if (condition3)
{
    //These statements would execute if the condition3 is true
}
.
.
else
{
    //These statements would execute if all the conditions return false.
}
```

## Example of else-if statement

Let's take the same example that we have seen above while discussing nested if-else. We will rewrite the same program using else-if statements.

```
#include <stdio.h>
int main()
{
    int var1, var2;
    printf("Input the value of var1:");
    scanf("%d", &var1);
    printf("Input the value of var2:");
    scanf("%d",&var2);
    if (var1 !=var2)
    {
        printf("var1 is not equal to var2\n");
    }
    else if (var1 > var2)
    {
        printf("var1 is greater than var2\n");
    }
    else if (var2 > var1)
    {
        printf("var2 is greater than var1\n");
    }
    else
    {
        printf("var1 is equal to var2\n");
    }
}
```

```
    return 0;  
}
```

Output:

```
Input the value of var1:12
```

```
Input the value of var2:21
```

```
var1 is not equal to var2
```

As you can see that only the statements inside the body of “if” are executed. This is because in this statement as soon as a condition is satisfied, the statements inside that block are executed and rest of the blocks are ignored.

### Important Points:

1. **else** and **else-if** are optional statements, a program having only “**if**” statement would run fine.
2. **else** and **else-if** cannot be used without the “**if**”.
3. There can be any number of **else-if** statement in a **if else-if** block.
4. If none of the conditions are met then the statements in **else** block gets executed.
5. Just like relational operators, we can also use logical operators such as **AND** (&&), **OR**(||) and **NOT**(!).