If else Statements

In programming, we often encounter situations where certain blocks of code should only execute if particular conditions are met. Java provides various control statements to manage the flow of execution based on these conditions.

Java's primary selection statements include:

- if
- if-else
- Nested-if
- if-else-if
- Switch-case
- Jump statements (break, continue, return)

In this article, we will focus on the first four: if, if-else, if-else-if, and nested-if

1. if:

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.

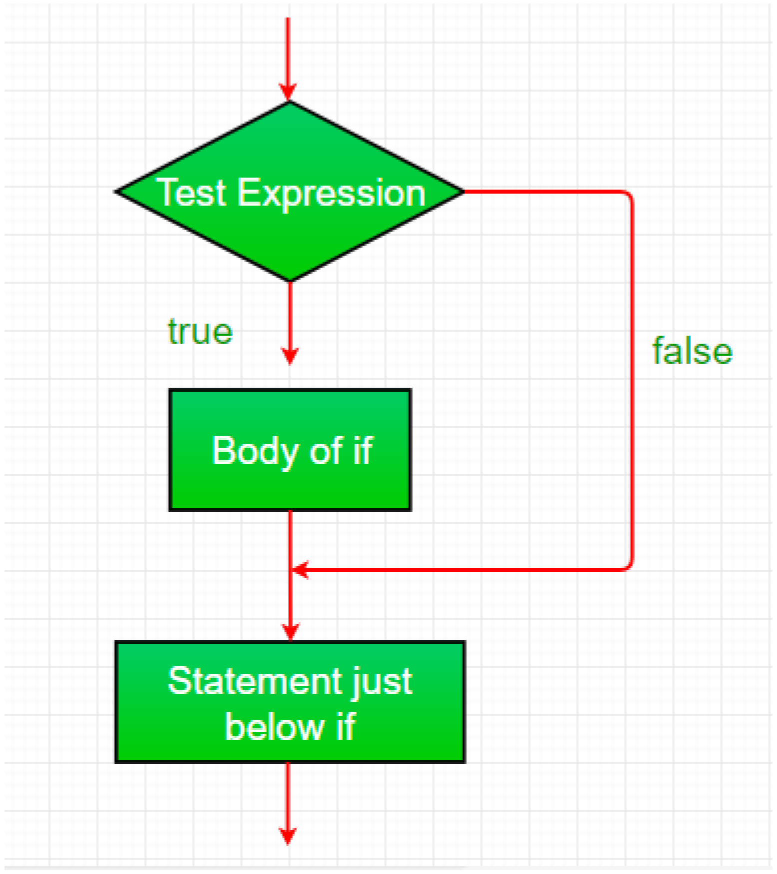
Syntax:

```
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 under it.
- If we do not provide the curly braces '{' and '}' after if(condition) then by default if statement will consider the immediate one statement to be inside its block. For example,

```
if(condition)
statement1;
statement2;

// Here if the condition is true, if block
// will consider only statement1 to be inside
// its block.
```



Example:

```
// Java program to illustrate If statement
class GfG {
    public static void main(String args[]) {
        int i = 10;

        if (i > 15)
            System.out.println("Greater than 15");

        // This statement will be executed
        // as if considers one statement by default
        System.out.println("I am not in if");
    }
}
```

Output

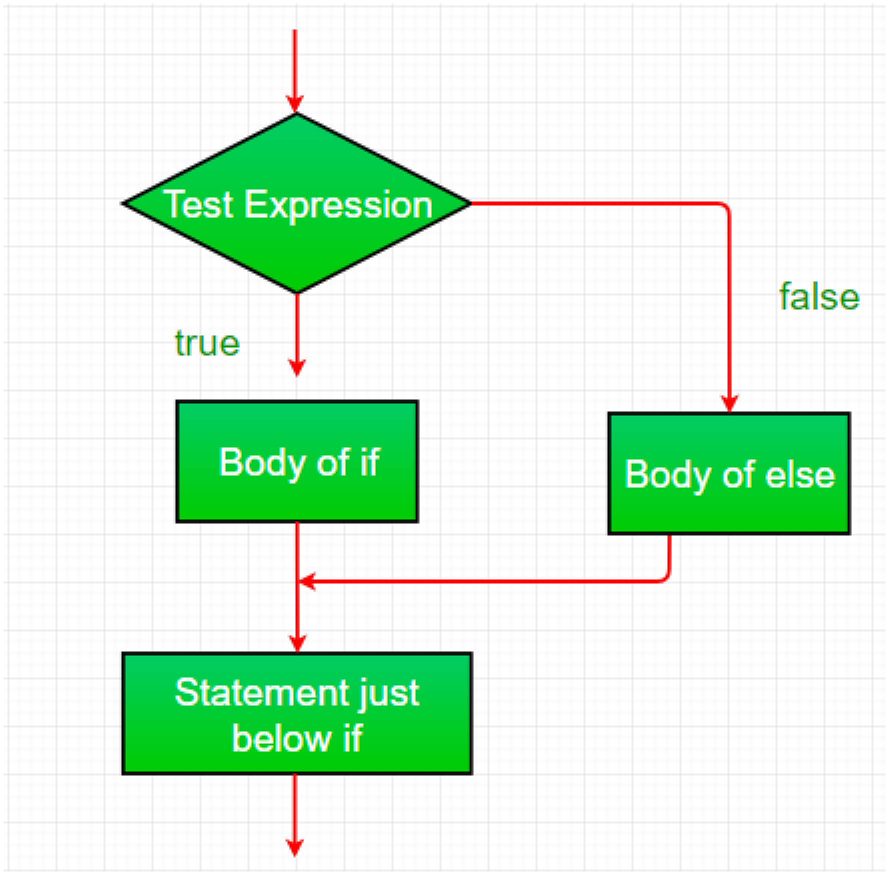
```
I am not in if
```

2. 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
}
```



Example:

```
// Java program to illustrate if-else statement
class GfG {
    public static void main(String args[]) {
        int i = 10;

        if (i < 15)
            System.out.println("i is smaller than 15");
        else
            System.out.println("i is greater than 15");
    }
}
```

Output

```
i is smaller than 15
```

Time Complexity: O(1)

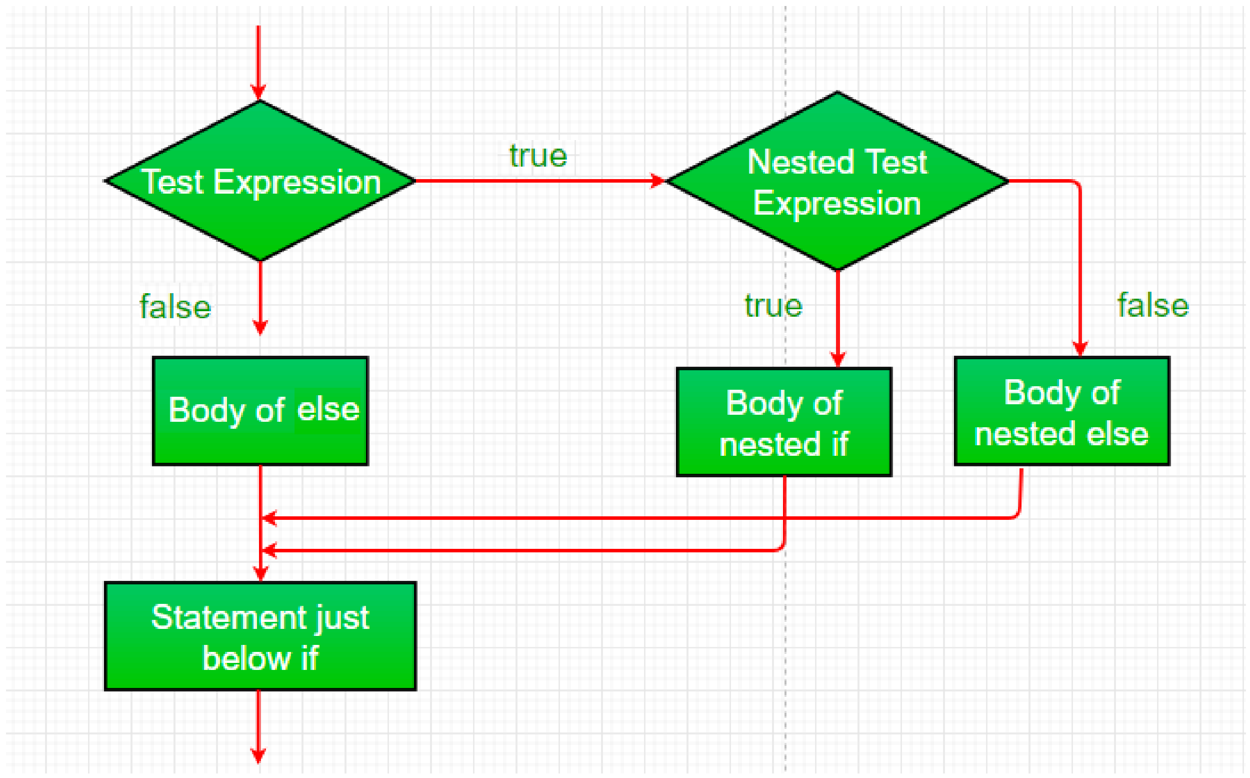
Auxiliary Space : O(1)

3. Nested-if:

A nested if is an if statement that is the target of another if or else. Nested if statements mean an if statement inside an if statement. Yes, java 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
}
}
```



Example:

```
// Java program to illustrate nested-if statement

class GfG {
    public static void main(String args[]) {
        int i = 10;

        if (i == 10 || i<15) {
            // First if statement
            if (i < 15)
                System.out.println("i is smaller than 15");

            // Nested - if statement
            // Will only be executed if statement above
            // it is true
            if (i < 12)
                System.out.println(
                    "i is smaller than 12 too");
        }
        else {
                System.out.println("i is greater than 15");
        }
    }
}
```

```
Output
i is smaller than 15
i is smaller than 12 too
```

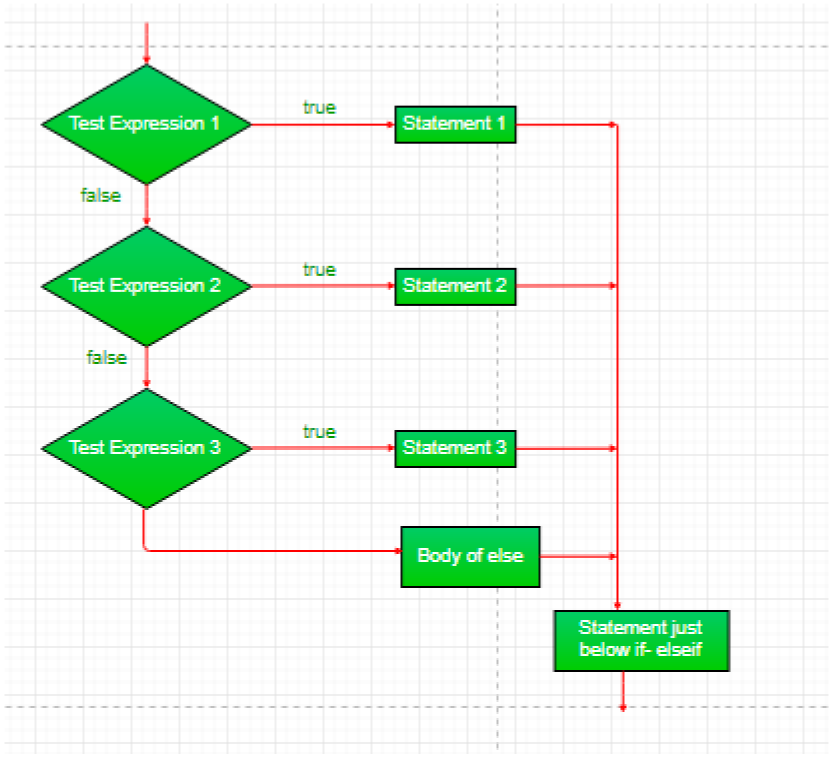
Time Complexity: O(1)

Auxiliary Space: O(1)

4. if-else-if ladder:

Here, a user can decide among multiple options. The 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 ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.

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



Example:

```
// Java program to illustrate if-else-if ladder

class GfG {
    public static void main(String args[]) {
        int i = 20;

        if (i == 10)
            System.out.println("i is 10");
        else if (i == 15)
            System.out.println("i is 15");
        else if (i == 20)
            System.out.println("i is 20");
        else
            System.out.println("i is not present");
    }
}
```

Output

```
i is 20
```

Time Complexity: O(1)

Auxiliary Space: O(1)