Java If-else Statement (Introduction)

Syntax:

```
if(condition) {
   Statement(s);       //This block will execute only if the condition is true
}
else {
   Statement(s);              //This block will execute if the condition is false
}
```

Example:

```
public class IfElseExample {
    public static void main(String args[]){
        int num=9;
            if( num < 0){
           System.out.println("Negative Number");
            }
           else{
              System.out.println("Positive Number");
           }
        }
   }
```

Output of this if else program in java will be:

```
Positive Number
```

Java If Statement

We use the if statement in java when we want to execute a block of code when a condition is true.

Syntax:

```
if (condition) {
   statement(s);                    // This code will execute if the condition is true
}
```

The statement will execute only if the condition is **true**, if the condition is **false** the code written inside the if block will be ignored, and the else block if present will be executed.

Example:

```
public class IfStatementInJavaExample {
   public static void main(String args[]){
      int num=400;
      if( num < 1000 ){
          /* The statement inside this block will only execute if the above condition is tr
          System.out.println("The given number is less than 1000");
      }
   }
}
```

Output of this if else program in java will be:

```
The given number is less than 1000
```

Java If with String

We can use string in if condition.

Example:

```
public class IfStatementWithStringExample {
   public static void main(String args[]){
      String str1="fifty";
      String str2="fifty";
      if( str1.equals(str2) ){
         /* The statement inside this block will only execute,  if the above condition is
         System.out.println("The given number is 50");
      }
   }
}
```

Output of this if else program in java will be:

```
The given number is 50
```

> Note: In java, we use **equals()** to compare values instead of **==** because **==** only checks if both objects refer to the same memory location, whereas the equals() method compares the values in the object.

Java if-else (if-then-else) Statement

We can write an if-else program in java, in which if the condition is **true**, the statement inside the if block will execute, and if the condition is **false**, the else block will execute.

Syntax:

```
if(condition) {
   Statement(s);        //This block will execute only if the condition is true
}
else {
   Statement(s);               //This block will execute if the condition is false
}
```

Example:

```
public class IfElseExample {
   public static void main(String args[]){
        int num=9;
        if( num % 2 == 0){
           System.out.println("Even Number");
        }
           else{
              System.out.println("Odd Number");
           }
        }
   }
```

The above if-else program in java uses an if-else statement to print Even or odd after checking it. **Output of this if else program in java will be:**

```
Odd Number
```

Java Nested if Statement

In java programming, if there is an if statement written inside another if block, then it is called a **nested if statement**.

Note: The inner if statement will be checked only if the outer if statement is true.

Syntax:

```
if(conditionOne) {
   StatementFirst(s);
   if(conditionTwo) {
      StatementSecond(s);
   }
}
```

Example:

```
public class NestedIf{
   public static void main(String args[]){
    int num=95;
        if( num > 50 ){
        // check if number is greater than 50
         System.out.println("number is greater than 50");
         if(num > 75){
             // check if number is greater than 75
                 System.out.println("number is greater than 75");
             }
        }
  }
}
```

Output of this if else program in java will be:

```
number is greater than 50
number is greater than 75
```

If else If Ladder Statement

In java programming language, we use the **if-else-if** statement when we want to check multiple conditions and want to execute a different statement depending on these conditions. In this, we have one if statement, one else statement and we can have multiple else if statements. This is known as the **if-else-if ladder**.

Syntax:

```
if(conditionOne) {
   statement(s);                          /*This block will execute if condtionOne is true
}
else if(conditionTwo) {
   statement(s);                /* This block will execute if conditionOne is not met and
}
else if(conditionThree) {
   statement(s);        /* This block will execute if conditionOne and conditionTwo is not
}
.
.
else {
   statement(s);        // This block will execute if all of the above condition is false.
}
```

Note: In an if-else-if statement as soon as the condition is true the set of code inside that block will get executed and every other else if and else block will be ignored, if no condition is true then the else block will get executed.

```java
public class IfElseIfExample{
    public static void main(String[] args) {
        int marks =  75;
        if (marks > 90){
            System.out.println("Excellent");
        }
        else if (marks > 80){
            System.out.println("Very Good");
        }
        else if (marks > 70){
            System.out.println("Good");
        }
        else if (marks > 60){
            System.out.println("Average");
        }
        else if (marks > 50){
            System.out.println("poor");
        }
        // default statement
        else {
            System.out.println("Fail");
        }
    }
}
```

Here in this example, the third condition is **true**, therefore it will print **"Good"** and everything else will be ignored.

Output of this if else program in java will be:

```
Good
```

Using Ternary Operator

Java programming language also supports a ternary operator. The ternary operator can work as an alternative for an if-else statement. Using the ternary operator, we can do the same work that we do with the help of an if-else statement, but it helps to shorten the codebase. The ternary operator is represented using "?:"

Syntax:

```
result = Expression1 ? Statement1: Statement2
```

Here if Expression1 is true then Statement1 will be executed and the resultant value will be stored in the result variable, and if Expression1 is false then Statement2 is executed and the resultant value will be stored in the result variable.

Example:

```
class TernaryOperatorInJava {
        public static void main(String[] args)
        {
                int num1 = 10, num2 = 20, largest;

    // ternary operator is used to compare num1 and num2
    // if num1 is greater than num2, then largest will be assigned num1
    // otherwise, largest will be assigned num2
                largest= (num1 > num2) ? num1 : num2;

    // printing the value of largest variable
                System.out.println("Largest is = " + largest);
        }
}
```

Output of this if else program in java will be:

```
Largest is = 20
```

Here in this example, it checks if **num1 > num2** so as num1 is less than num2 and the condition evaluates to false therefore num2 value will be stored in variable largest. Hence **Largest is = 20** is printed on the screen.