Java if else Statement

In Java, if statement is used for testing the conditions. The condition matches the statement it returns true else it returns false.

For example, if we want to create a program to test positive integers then we have to test the integer whether it is greater than zero or not.

In this scenario, if statement is helpful.

There are four types of if statement in Java:

1. if statement

2. if-else statement

3. if-else-if ladder

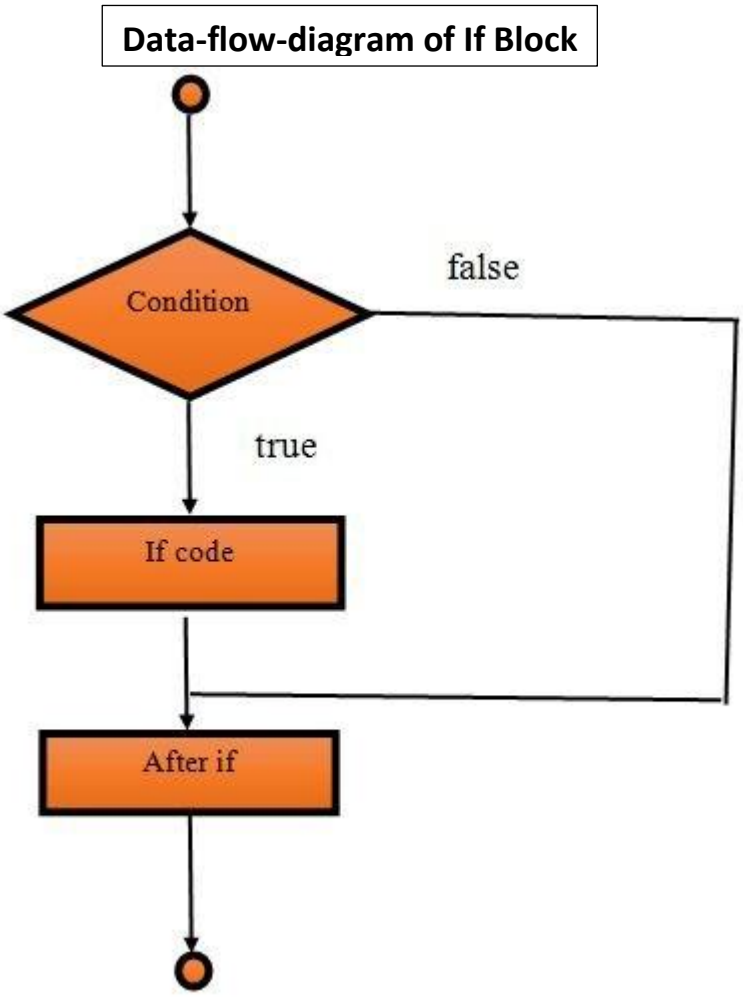
4. nested if statement

1. if Statement

The if statement is a single conditional based statement that executes only if the provided condition is true. Syntax of if:

```
if(condition)
{
        //code
}
```

We can understand flow of if state
that code written inside the if will



- Example of if:

In this example, we are testing student's marks. If the marks are greater than 65 then student will get first division.

```
class IfDemo1 {
        public static void main (String[] args) {
                int marks=70;
                if(marks > 65)
                {
                        System.out.print("First division");
                }
        }
}
```

2. if-else Statement

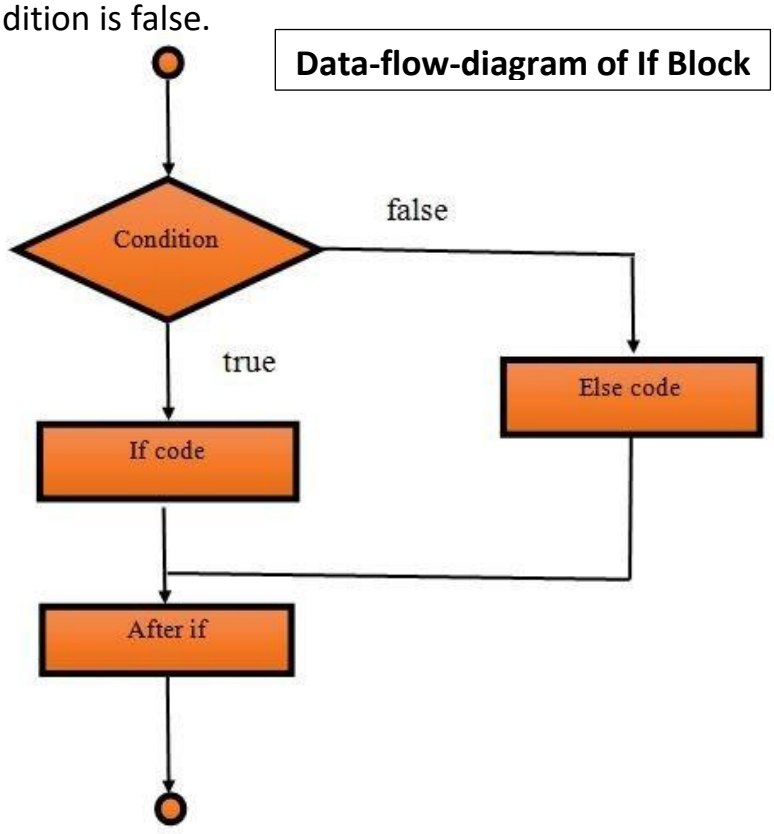
The if-else statement is used for testing condition. If the condition is true, if block executes otherwise else block executes.

It is useful in the scenario when we want to perform some operation based on the false result.

The else block executes only when condition is false.

- Syntax of if else:

```
if(condition)
{
        //code for true
}
else
{
        //code for false
}
```



In this block diagram, we can see that when condition is true, if block executes otherwise else block executes.

- Example of if else:

In this example, we are testing student marks, if marks is greater than 65 then if block executes otherwise else block executes.

```
class IfElseDemo1 {
	public static void main(String[] args)
	{
		int marks=50;
		if(marks > 65)
		{
			System.out.print("First division");
		}
		else
		{
			System.out.print("Second division");
		}
	}
}
```

3. if-else-if ladder Statement

In Java, the if-else-if ladder statement is used for testing conditions. It is used for testing one condition from multiple statements.

When we have multiple conditions to execute then it is recommend to use if-else-if ladder.

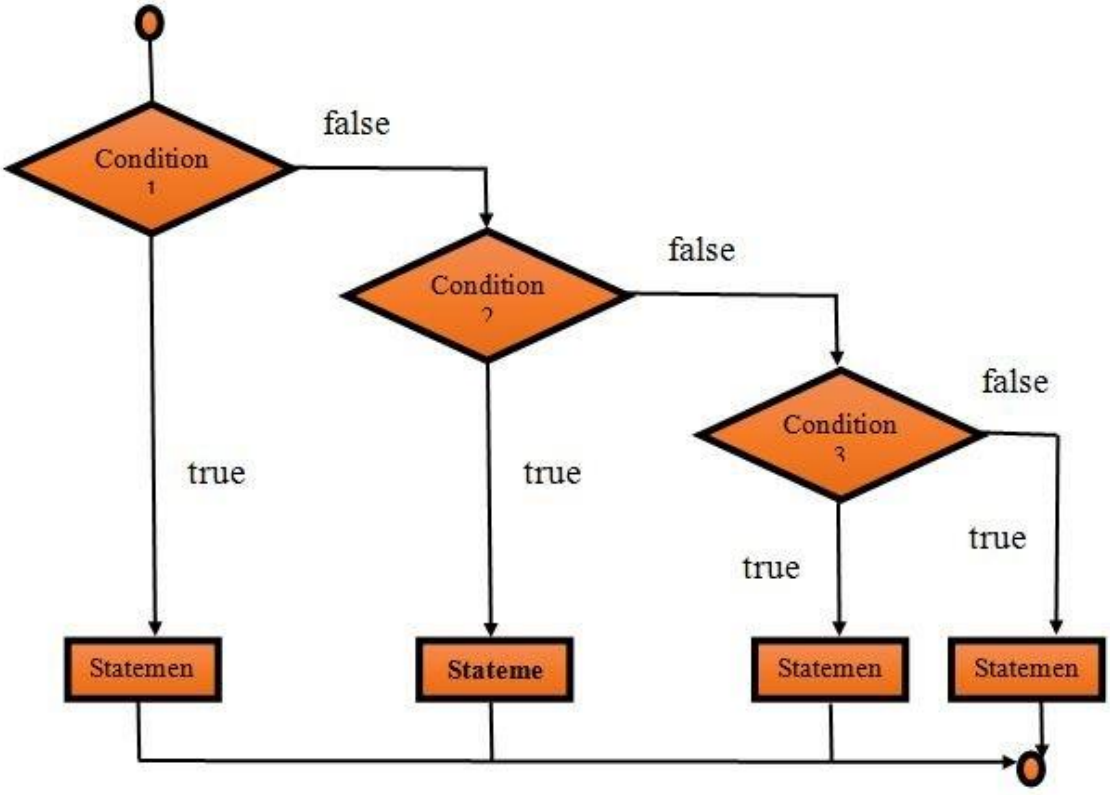
- Syntax:

```
if(condition1)
{
	//code for if condition1 is true
}
else if(condition2)
```

```
{
        //code for if condition2 is true
}
else if(condition3)
{
        //code for if condition3 is true
}
...
else
{
        //code for all the false conditions
}
```

It contains multiple conditions and execute if any condition is true otherwise executes else block.

- Data-flow-diagram of If Else If Block



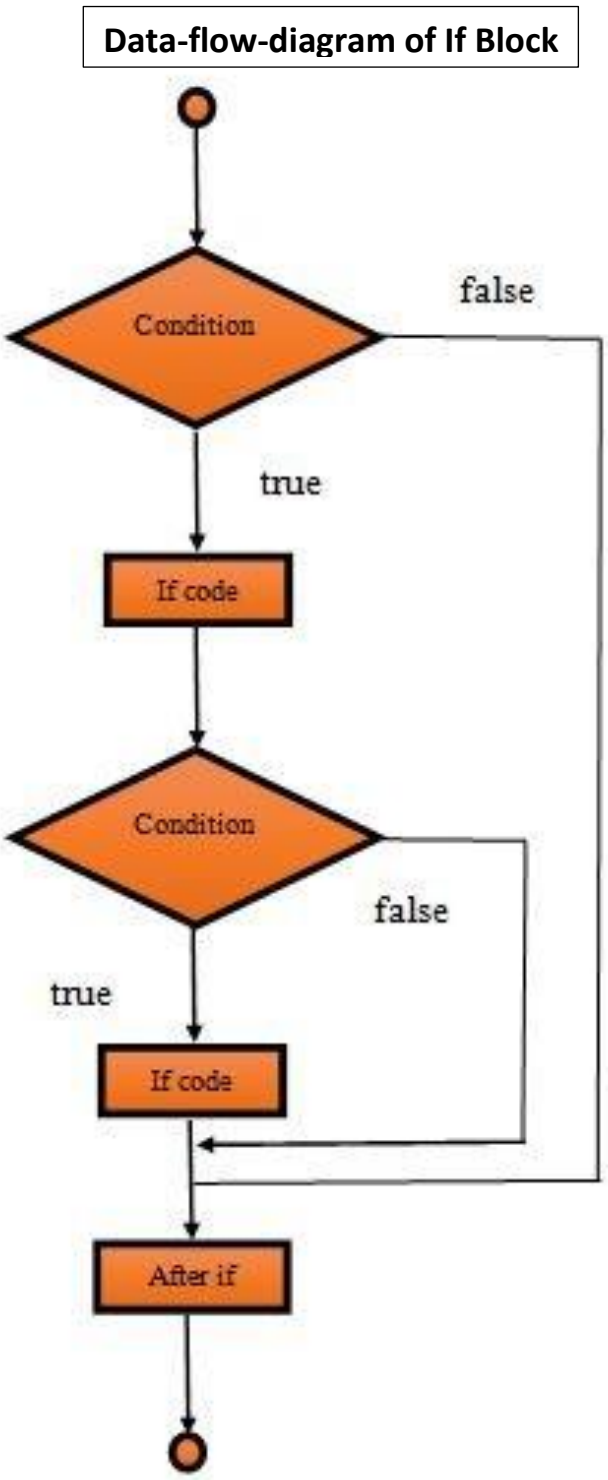
- Example:

Here, we are testing student marks and displaying result based on the obtained marks. If marks are greater than 50 student gets his grades.

```
class IfElseIfDemo1 {
	public static void main(String[] args) {
		int marks=75;
		if(marks<50){
			System.out.println("fail");
		}
		else if(marks>=50 && marks<60){
			System.out.println("D grade");
		}
		else if(marks>=60 && marks<70){
			System.out.println("C grade");
		}
		else if(marks>=70 && marks<80){
			System.out.println("B grade");
		}
		else if(marks>=80 && marks<90){
			System.out.println("A grade");
		}
		else if(marks>=90 && marks<100){
			System.out.println("A+ grade");
		}
		else{
			System.out.println("Invalid!");
		}
	}
}
```

4. Nested if statement

In Java, the Nested if statement is an if inside another if. In this, one if block is created inside another if block when the outer block is true then only the inner block is executed.



- Syntax:

```
if(condition)
{
        //statement
        if(condition)
        {
                //statement
        }
}
```

- Example:

```
class NestedIfDemo1 {
public static void main(String[] args)
{
        int age=25;
        int weight=70;
        if(age>=18) {
                if(weight>50)
                {
                        System.out.println("You are eligible");
                }
        }
}
}
```