Operators

Operators are symbols which performs the operations on variables or values.

Categories:

1. Arithmetic Operators (+, -, *, /, %)
2. Relational Operators (==, !=, >, <, >=, <=)
3. Logical Operators(&&, ||, !)

- Arithmetic Operators
  Used for basic mathematical operations.

```
public class ArithmeticExample {
    public static void main(String[] args) {
        int a = 30,b = 10;
        System.out.println("Addition= "+ (a+b));
        System.out.println("Subtraction= "+ (a-b));
        System.out.println("Multiplication= "+ (a*b));
        System.out.println("Division= "+ (a/b));
        System.out.println("Modulus= "+ (a%b));
    }
}
```

- Relational Operators
  - Used to compare the values

```
public class RelationalExample {
    public static void main(String[] args){
        int a =10,b=5;
        System.out.println("a == b ? ="+ (a==b));
        System.out.println("a != b ? ="+ (a != b));
        System.out.println("is a greater than b? "+
(a > b));
        System.out.println("a < b "+ (a < b));
    }
}
```

- Logical Operators
  - Used to combine multiple conditions

```
public class LogicalExample {
    public static void main(String[] args) {
        int age = 20;
        boolean hasId = true;

        // AND Operator
        // T T -> True
        System.out.println("Eligible = "+(age >= 18
&& hasId));

        // OR Operator
        // F F -> False
        System.out.println(age < 18 || hasId);

        //NOT Operator
        //!(T) -> false
        //!(F) -> true
        System.out.println(!(age >= 18));
        // !(age >= 18) -> !(T) -> False
    }
}
```

Control Statements:

Allows us to control the flow of execution.

If statement:

It is used to execute a block of a code if a condition evaluates to be true.

Syntax:

```
If(condition){
      //code to execute if the condition is true
}
```

```java
public class IfExample {
    public static void main(String[] args) {
        int num = 10;

        if(num > 0){
            System.out.println("The number is positive");
        }
    }
}
```

if-else statement:

it gives us alternative path of execution when the condition is false

```
if(condition){
      // code to be executed if the condition is true
} else {
      // code to be executed if the condition is false
}
```

```java
public class ifElseExample {

    public static void main(String[] args) {
        int num = -10;

        if(num > 0){
            System.out.println("The number is positive");
        }else{
            System.out.println("The number is negative");
        }
    }
}
```

if-else-if ladder:

- It is used to test multiple conditions

SYNTAX:

```
If(condition1){
      // code to be executed if the condition1 is true
} else if(condition2){
      // code to be executed if the condition2 is true
}else{
     // code to be executed if none of the conditions are true
}
```

```java
public class IfElseIfExample {
    public static void main(String[] args) {
        int marks = 70;

        if (marks >= 90){
            System.out.println("Grade A+");
        } else if (marks >= 80) {
            System.out.println("Grade A");
        } else if (marks >= 70) {
            System.out.println("Grade B");
        } else if (marks >=60) {
            System.out.println("Grade C");
        } else{
            System.out.println("FAIL");
        }
    }
}
```