

OPERATOR

Operator is a symbol that is used to perform operation according to user requirement.

- **Arithmetic operator(+,-,*,/)**
- **Relation operator(==,!=,>,<=<,>)**
- **Logical operator(&&,||,!)**
- **Increment & Decrement**

Using Switch & Arithmetic Op:

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {

        char operator;
        Double number1, number2, result;

        // create an object of Scanner class
        Scanner input = new Scanner(System.in);

        // ask users to enter operator
        System.out.println("Choose an operator: +, -, *, or /");
        operator = input.next().charAt(0);

        // ask users to enter numbers
        System.out.println("Enter first number");
        number1 = input.nextDouble();

        System.out.println("Enter second number");
        number2 = input.nextDouble();

        switch (operator) {

            // performs addition between numbers
            case '+':
                result = number1 + number2;
                System.out.println(number1 + " + " + number2 + " = " + result);
                break;

            // performs subtraction between numbers
            case '-':
                result = number1 - number2;
                System.out.println(number1 + " - " + number2 + " = " + result);
                break;

            // performs multiplication between numbers
```

```

        case '*':
            result = number1 * number2;
            System.out.println(number1 + " * " + number2 + " = " + result);
            break;

        // performs division between numbers
        case '/':
            result = number1 / number2;
            System.out.println(number1 + " / " + number2 + " = " + result);
            break;

        default:
            System.out.println("Invalid operator!");
            break;
    }

    input.close();
}
}

```

OUTPUT:- Choose an operator: +, -, *, or /
 *
 Enter first number
 3
 Enter second number
 9
 3.0 * 9.0 = 27.

Using logical Op:

```

class Main {
    public static void main(String[] args) {

        // && operator
        System.out.println((5 > 3) && (8 > 5)); // true
        System.out.println((5 > 3) && (8 < 5)); // false

        // || operator
        System.out.println((5 < 3) || (8 > 5)); // true
        System.out.println((5 > 3) || (8 < 5)); // true
        System.out.println((5 < 3) || (8 < 5)); // false

        // ! operator
        System.out.println(!(5 == 3)); // true
        System.out.println(!(5 > 3)); // false
    }
}

```

OUTPUT:- true
 false
 true
 true
 false
 true
 false

Relational Op:

```
class Main {
    public static void main(String[] args) {

        // create variables
        int a = 7, b = 11;

        // value of a and b
        System.out.println("a is " + a + " and b is " + b);

        // == operator
        System.out.println(a == b); // false

        // != operator
        System.out.println(a != b); // true

        // > operator
        System.out.println(a > b); // false

        // < operator
        System.out.println(a < b); // true

        // >= operator
        System.out.println(a >= b); // false

        // <= operator
        System.out.println(a <= b); // true
    }
}
OUTPUT:- a is 7 and b is 11
false
true
false
true
false
true
```

Increment and Decrement Op:

```
class Main {
    public static void main(String[] args) {

        // declare variables
        int a = 12, b = 12;
        int result1, result2;

        // original value
        System.out.println("Value of a: " + a);

        // increment operator
        result1 = ++a;
        System.out.println("After increment: " + result1);

        System.out.println("Value of b: " + b);
```

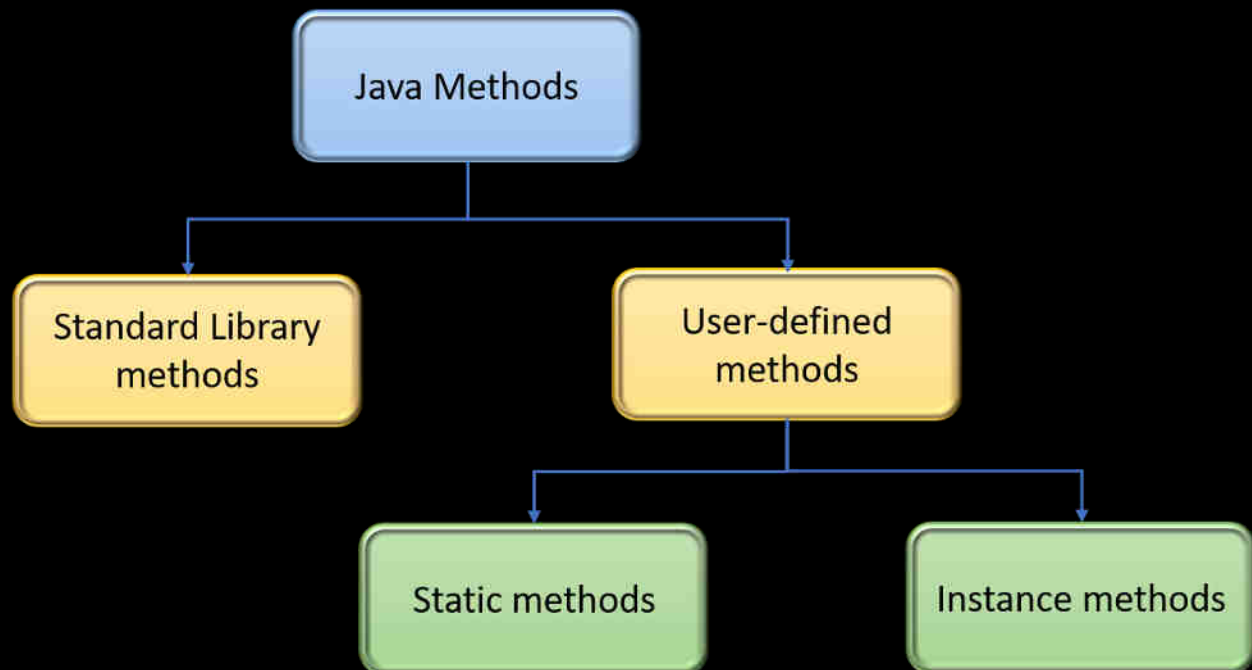
```

        // decrement operator
        result2 = --b;
        System.out.println("After decrement: " + result2);
    }
}
OUTPUT:- Value of a: 12
After increment: 13
Value of b: 12
After decrement: 11

```

Method Or Function:

- Method is a group/block of code which take input from the user processed it and give output.
- Method runs only when it called
- Code reusability



Syntax:-

```

return-type methodName(){
    //code
}

```

Program:-

```

class Main {
    static int sum(int num1, int num2) {
        return num1 + num2;
    }
}

```

```
public static void main(String[] args) {  
    int r = sum(10, 20);  
    System.out.print("Addition: " + r);  
}  
}
```

OUTPUT:- Addition: 30