

The **non-primitive** data types are a little advanced concepts which we will cover once we have mastered the primitives and are well versed with the programming principles of Java.

Topic: Operators in Java

Operators in Java can be classified into 6 types:

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Unary Operators
6. Bitwise Operators

1. Java Arithmetic operators:

Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. The following table lists the arithmetic operators:

Assume integer variable num1 holds 20 and variable num2 holds 30, then:

Operator	Description and Example
+ Addition	Adds values on either side of the operator. <i>Example: num1 + num2 will give 50</i>
- Subtraction	Subtracts right hand operand from left hand operand <i>Example: num1 - num2 will give -10</i>
* Multiplication	Multiplies values on either side of the operator <i>Example: num1 * num2 will give 600</i>
/ Division	Divides left hand operand by right hand operand <i>Example: num1 / num2 will give 0.67</i>
% Modulus	Divides left hand operand by right operand and returns remainder <i>Example: num2 % num1 will give 10</i>
++ Increment	Increases the value of operand by 1 <i>Example: num2++ gives 31</i>
-- Decrement	Decreases the value of operand by 1 <i>Example: num1-- gives 19</i>

Example:

```
class Main {  
    public static void main(String[] args) {  
  
        // declare variables p and q  
        int p = 20, q = 10;  
        int result;  
  
        // addition operator  
        result=p+q;  
        System.out.println(result);  
  
        // subtraction operator  
        System.out.println(p - q);  
  
        // we can directly perform subtraction in print statement,no need to  
        // use result variable here  
  
        // multiplication operator  
        System.out.println(p * q);  
  
        // division operator  
        System.out.println(p / q);  
  
        // modulo operator  
        System.out.println(p % q);  
    }  
}
```

Output:

```
30  
10  
200  
2  
0
```

Increment and Decrement Operators

1. PreIncrement(++a)
2. PostIncrement(a++)
3. PreDecrement(--a)
4. PostDecrement(a--)

Example:

```
class Main {  
    public static void main(String[] args) {  
        int a = 5, int b = 6;  
        int c = a++; //post increment  
        int d = ++a; //pre increment  
        int e = b--; //post decrement  
        int f = -b; // pre decrement  
  
        System.out.println(c);  
        System.out.println(d);  
        System.out.println(e);  
        System.out.println(f);  
    }  
}
```

Output:

```
5  
7  
6  
4
```