

## Arithmetic operators

Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups:

- Arithmetic Operators
- Relational Operators
- Bitwise Operators
- Logical Operators
- Assignment Operators
- Misc Operators

### The Arithmetic Operators:

Operator	Description	Example
+	Addition	3 + 4
–	Subtraction	5 – 7
*	Multiplication	5 * 5
÷	Division	14 ÷ 7
%	Modulus	20 % 7

Each operator takes two operands, one on either side of the operator. The subtraction operator (–) can also be used to negate a single operand.

Integer division results in an integer. Because integers don't have decimal fractions, any remainder is ignored. The expression  $31 \div 9$ , for example, results in 3 (9 goes into 31 only 3 times).

Modulus (%) gives the remainder once the operands have been evenly divided. For example, 31

% 9 results in 4 because 9 goes into 31 three times, with 4 left over.

Note that, for integers, the result type of most operations is an int or a long, regardless of the original type of the operands. Large results are of type long; all others are int. Arithmetic wherein one operand is an integer and another is a floating point results in a floating-point result. (If you're interested in the details of how Java promotes and converts numeric types from one type to another.

### For Example:

The following simple example program demonstrates the arithmetic operators.

```
public class Test{

    public static void main(String args[])

    {

        int a =10;

        int b =20;

        int c =25;

        int d =25;

        System.out.println("a + b = "+(a + b));

        System.out.println("a - b = "+(a - b));

        System.out.println("a * b = "+(a * b));

        System.out.println("b / a = "+(b / a));

        System.out.println("b % a = "+(b % a));

        System.out.println("c % a = "+(c % a));

        System.out.println("a++ = "+(a++));
```

```
System.out.println("b-- = "+(a--));
```

```
// Check the difference in d++ and ++d
```

```
System.out.println("d++ = "+(d++));
```

```
System.out.println("++d = "+(++d));
```

```
}
```

```
}
```

### Out Put Screen

a + b =30

a - b =-10

a \* b =200

b / a =2

b % a =0

c % a =5

a++=10

b--=11

d++=25

++d =27