Operators

The main objective of any computer program is to be functional. Being functional means to carry out certain operations in it with respect to various variables and values present in it. This overall process is guided via the use of **Operators**. So, we must be familiar with various operators and their usages in the computer programs. For example; + is an operator used for addition whereas – is an operator used for subtraction.

Here, in this chapter we will be learning about various operators available in JAVA along with their syntax and how to use them via the demonstration of certain examples.

In JAVA, operators can be classified into following types;

1. Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on variables and data. For example;

```
a+b;
a-b;
```

Here, + operator is used for adding two variables a and b whereas – operator is used to perform subtraction. There are other arithmetic operators as well which are as;

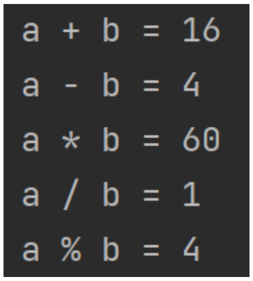
Operator	**Operation**
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Operation (Remainder after division)

Now let’s see an example as;

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

      // declare variables
      int a = 10, b = 6;

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

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

```
        // multiplication operator
        System.out.println("a * b = " + (a * b));

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

        // modulo operator
        System.out.println("a % b = " + (a % b));
    }
}
```

Output



```
a + b = 16
a - b = 4
a * b = 60
a / b = 1
a % b = 4
```

2. Assignment Operators

Likewise, assignment operators in JAVA are used to assign values to variables. For example;

```
int a ;
a= 100;
```

Here, = is the assignment operator. It assigns the value on its right to the variable on its left. That is 100 is assigned to the variable a.

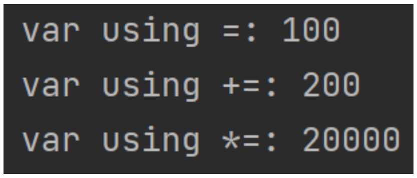
Now let us learn about other assignment operators available in JAVA.

Operator	Example	Equivalent to
=	a = b;	a = b;
+=	a += b;	a = a + b;
-=	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;
%=	a %= b;	a = a % b;

Now let’s see an example as;

```
class Main {
    public static void main(String[] args) {
        // create variables
        int a = 100;
        int var;

        // assign value using =
        var = a;
        System.out.println("var using =: " + var);

        // assign value using =+
        var += a;
        System.out.println("var using +=: " + var);

        // assign value using =*
        var *= a;
        System.out.println("var using *=: " + var);
    }
}
```

Output



```
var using =: 100
var using +=: 200
var using *=: 20000
```

3. Relational Operators

Similarly, Relational Operators are used to check the relationship between two operands. For example;

```
// check if a is less than b
        a < b;
```

Here, < operator is the relational operator. It checks if the value of a is less that b or not. It returns either true or false based upon the evaluation done. Mostly, relational operators are used in decision making statements and iterations.

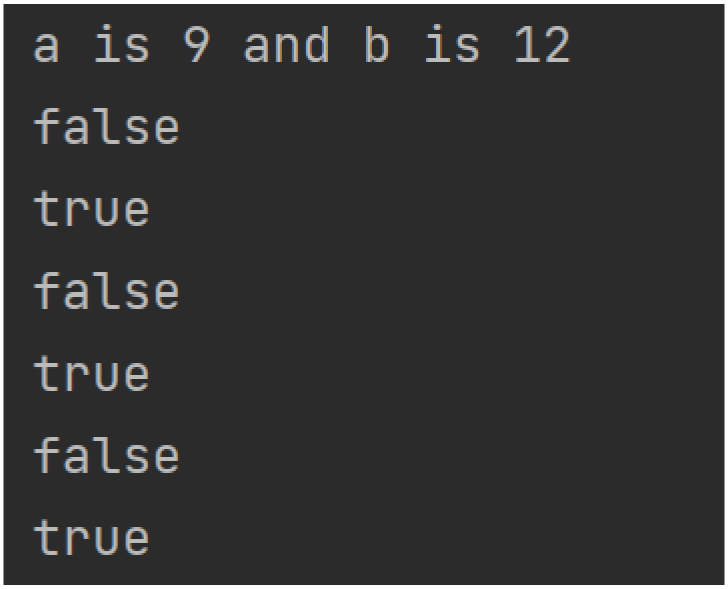
Operator	**Description**	**Example**
==	Is Equal To	3 == 5 returns **false**
!=	Not Equal To	3 != 5 returns **true**
>	Greater Than	3 > 5 returns **false**
<	Less Than	3 < 5 returns **true**
>=	Greater Than or Equal To	3 >= 5 returns **false**

<=	Less Than or Equal To	3 <= 5 returns **true**

Now let’s see an example as;

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

      // create variables
      int a = 9, b = 12;

      // 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 9 and b is 12
false
true
false
true
false
true
```

4. Logical Operators

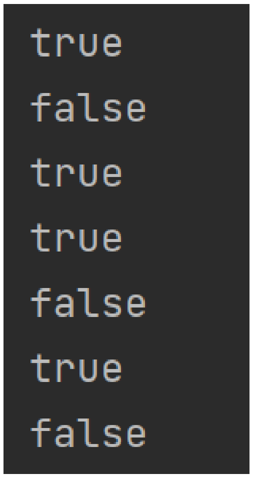
In the same way, Logical operators are used to check whether an expression is true or false. They are used during decision making.

Operator	**Example**	**Meaning**
&& (Logical AND)	expression1 **&&** expression2	true only if both expression1 and expression2 are true
|| (Logical OR)	expression1 || expression2	true if either expression1 or expression2 is true
! (Logical NOT)	**!**expression	true if expression is false and vice versa

Now let’s see an example as;

```
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
```

5. Unary Operators

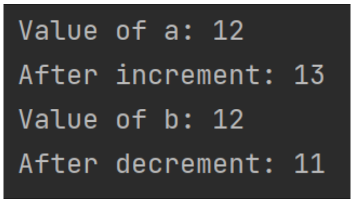
Moreover, Unary Operators are used with only one operand. For example ++ is a unary operator that increases the value of a variable by 1. For example; ++6 will return 7.

Operator	Meaning
+	**Unary plus**: not necessary to use since numbers are positive without using it
-	**Unary minus**: inverts the sign of an expression
++	**Increment operator**: increments value by 1
--	**Decrement operator**: decrements value by 1
!	**Logical complement operator**: inverts the value of a boolean

Now let’s see an example as;

```
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
```

6. Ternary Operator

The ternary operator (conditional operator) is shorthand for the if-then-else statement. It is statement as well as operator in itself.

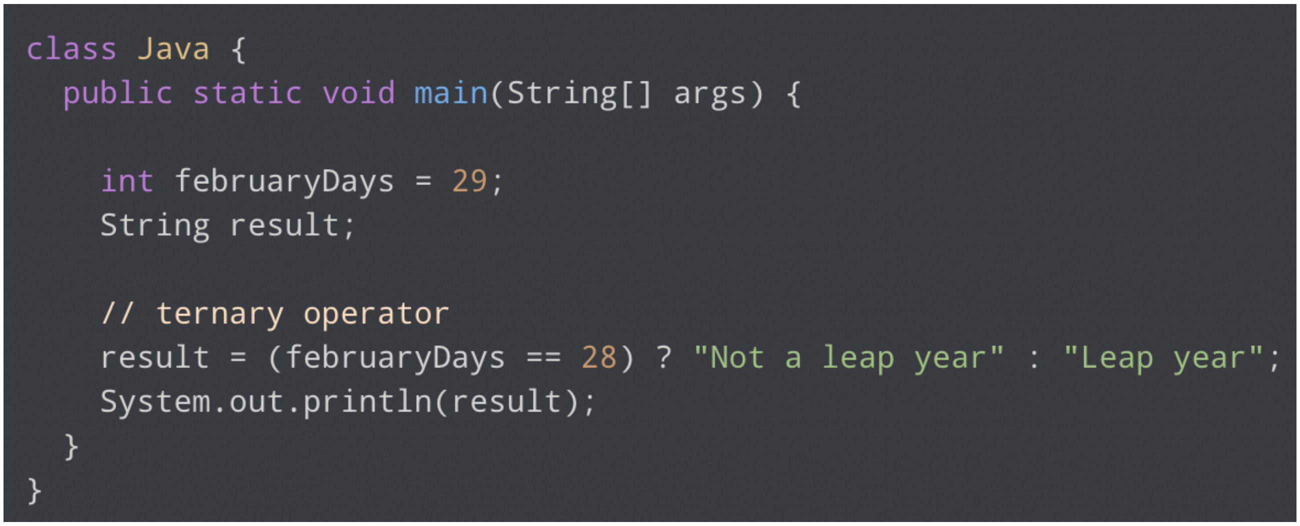
For example;

variable = Expression ? expression1 : expression2

Here's how it works.

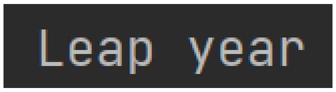
- if the Expression is true, expression1 is assigned to the variable.
- if the Expresson is false, expresson2 is assigned to the variable.

Now let's see an example as;



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

    int februaryDays = 29;
    String result;

    // ternary operator
    result = (februaryDays == 28) ? "Not a leap year" : "Leap year";
    System.out.println(result);
  }
}
```

Output



```
Leap year
```

Task to Do:

1. **Write a program to check whether a person can cast a vote or not. Condition is you must be above 18 years to vote.**
2. **Write a program to calculate SI, Area of triangle and Volume of Cube and Cuboid.**
3. **Write ternary operator for the question no. 1.**
4. **Write a program to calculate the total marks of four subjects of a student and the total percentage secured. And use following conditions to generate the final result;**
   a. **If equal to or more than 70 -> First Class**
   b. **If more than 59 -> Upper second Class**
   c. **If more than 49 -> Second class**
   d. **If more than 39 -> Third class and if below than 40 the result is fail.**