

Guide to Java operators with examples and Output.

1. Arithmetic Operators

Used for mathematical calculations.

Operator	Meaning
----------	---------

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

Example

```
public class ArithmeticExample {  
  
    public static void main(String[] args) {  
  
        int a = 20;  
        int b = 6;  
  
        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("Remainder: " + (a % b));  
    }  
}
```

Output

```
Addition: 26  
Subtraction: 14  
Multiplication: 120  
Division: 3  
Remainder: 2
```

Note: Since a and b are int, 20 / 6 gives 3, not 3.333.

2. Assignment Operators

Used to assign values to variables.

```
public class AssignmentExample {  
  
    public static void main(String[] args) {  
  
        int a = 10;  
  
        a += 5;  
        System.out.println("a += 5: " + a);  
  
        a -= 3;  
        System.out.println("a -= 3: " + a);  
  
        a *= 2;  
        System.out.println("a *= 2: " + a);  
  
        a /= 4;  
        System.out.println("a /= 4: " + a);  
    }  
}
```

Output

```
a += 5: 15  
a -= 3: 12  
a *= 2: 24  
a /= 4: 6
```

For example:

```
a += 5;
```

means:

```
a = a + 5;
```

3. Relational / Comparison Operators

Used to compare two values.

Operator	Meaning
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal

Example

```
public class ComparisonExample {  
  
    public static void main(String[] args) {  
  
        int a = 10;  
        int b = 20;  
  
        System.out.println(a == b);  
        System.out.println(a != b);  
        System.out.println(a > b);  
        System.out.println(a < b);  
        System.out.println(a >= b);  
        System.out.println(a <= b);  
    }  
}
```

Output

```
false  
true  
false  
true  
false  
true
```

4. Logical Operators

Used with conditions.

Operator Meaning

&&	AND
&	
!	NOT

Example

```
public class LogicalExample {  
  
    public static void main(String[] args) {  
  
        int age = 25;
```

```
        System.out.println(age >= 18 && age <= 60);
        System.out.println(age < 18 || age > 60);
        System.out.println(!(age >= 18));
    }
}
```

Output

```
true
false
false
```

Understanding &&

Both conditions must be true:

```
age >= 18 && age <= 60
```

Understanding ||

At least one condition must be true:

```
age < 18 || age > 60
```

Understanding !

Changes true to false and false to true:

```
!(age >= 18)
```

5. Increment and Decrement Operators

Operator	Meaning
++	Increase by 1
--	Decrease by 1

Example

```
public class IncrementExample {

    public static void main(String[] args) {

        int a = 10;

        a++;
        System.out.println("After ++: " + a);

        a--;
        System.out.println("After --: " + a);
    }
}
```

Output

```
After ++: 11
After --: 10
```

6. Ternary Operator

The ternary operator is a short way of writing if-else.

Syntax:

condition ? value1 : value2;

Example

```
public class TernaryExample {  
  
    public static void main(String[] args) {  
  
        int age = 20;  
  
        String result = (age >= 18) ? "Adult" : "Minor";  
  
        System.out.println(result);  
    }  
}
```

Output

Adult

This:

String result = (age >= 18) ? "Adult" : "Minor";

is similar to:

```
if (age >= 18) {  
    result = "Adult";  
} else {  
    result = "Minor";  
}
```