Java Operators

What is an Operator in Java?

Operator = A symbol that does some work on variables or values.
Example: + , - , * , / , = , > , <

1. Arithmetic Operators

Used for mathematical calculations.

Operator	Meaning	Example	Result
+	Add	5 + 2	7
-	Subtract	5 - 2	3
*	Multiply	5 * 2	10
/	Divide	10 / 2	5
%	Modulus (remainder)	10 % 3	1

Example:

```
int a = 10;
int b = 3;

System.out.println(a + b); // 13
System.out.println(a % b); // 1
```

2. Relational (Comparison) Operators

Used to compare two values. Result is **true or false**.

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

Example:

```
int x = 10, y = 5;
System.out.println(x > y); // true
System.out.println(x == y); // false
```

3. Logical Operators

Used with conditions (mostly in if, loops).

Operator	Meaning	Example
&&	AND	(a > b && a > c)
||	||	(a>b || a>c)
!	NOT	!(a > b)

Example:

```
int age = 20;
System.out.println(age > 18 && age < 60); // true
```

4. Assignment Operators

Used to assign values.

Operator	Meaning	Example
=	Assign	a = 10
+=	a = a + 5	a += 5
-=	a = a - 5	a -= 5
*=	a = a * 5	a *= 5
/=	a = a / 5	a /= 5
%=	a = a % 5	a %= 5

Example:

```
int a = 10;
a += 5; // a = 15
```

5. Unary Operators

Work on one variable only.

Operator	Meaning
+	Positive
-	Negative
++	Increment (increase by 1)
--	Decrement (decrease by 1)
!	Logical NOT

Example:

```
int a = 10;
a++; // 11
a--; // 10
```

6. Bitwise Operators

Work on binary bits.

Operator	Meaning
&	Bitwise AND
`	`
^	XOR
~	Bitwise NOT
<<	Left shift
>>	Right shift
>>>	Unsigned right shift

Example:

```
int a = 5; // 0101
int b = 3; // 0011
System.out.println(a & b); // 1
```

7. Ternary Operator (Shortcut of if-else)

Format:

```
condition ? value1 : value2;
```

Example:

```
int age = 20;
String result = (age >= 18) ? "Adult" : "Child";
System.out.println(result);
```

8. Special Operators

1.instanceof

Checks object type.

```
String s = "Hello";
System.out.println(s instanceof String); // true
```

2.Dot Operator .

Used to access methods and variables.

```
System.out.println();
```

3.new Operator

Used to create objects.

```
Scanner sc = new Scanner(System.in);
```

Important Notes

- Arithmetic → math
- Relational → compare
- Logical → combine conditions
- Assignment → store value
- Unary → ++, --
- Bitwise → binary operations
- Ternary → short if-else

What is Shift Operator?

Shift operator **moves bits left or right**.

Think like:
Bits = people standing in a line
Shift = move everyone left or right

1. Left Shift <<

Moves bits to the **left**
Adds **0 on the right side**

Easy Rule:

$a << n = a \times 2^{n}$

Example:

```
int a = 5;
System.out.println(a << 1);
```

Result:

5 << 1 = 10

Means multiply by 2

2. Right Shift >>

Moves bits to the **right**
Keeps the sign (positive or negative)

Easy Rule:

a >> n = a ÷ 2^n

Example:

```
int a = 10;
System.out.println(a >> 1);
```

Result:

10 >> 1 = 5

Means divide by 2

3. Unsigned Right Shift >>>

Moves bits to the **right**
Always puts **0 on left side**
Ignores negative sign

Example:

```
int a = -10;
System.out.println(a >> 1);
System.out.println(a >>> 1);
```

Output:

```
>>  = negative number
>>> = big positive number
```

Operator	Meaning	Easy Trick
<<	Left shift	Multiply by 2
>>	Right shift	Divide by 2
>>>	Unsigned right shift	Divide but remove negative sign

```
int money = 8;

System.out.println(money << 1); // 16 (double)
System.out.println(money >> 1); // 4 (half)
```

(Remember This)

<< increases value
>> decreases value
>>> removes sign effect