In Java, operators are special symbols or keywords that are used to perform operations on variables and values.

**Types of operators**

**1. Arithmetic Operators**

These operators are used to perform mathematical operations like addition, subtraction, multiplication, division, and modulus.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| + | Addition | a + b |
| - | Subtraction | a – b |
| \* | Multiplication | a \* b |
| / | Division | a / b |
| % | Modulus (Remainder) | a % b |

Example:

java

Copy

int a = 10, b = 5;

System.out.println(a + b); // Output: 15

System.out.println(a - b); // Output: 5

System.out.println(a \* b); // Output: 50

System.out.println(a / b); // Output: 2

System.out.println(a % b); // Output: 0

**2. Relational (Comparison) Operators**

These operators are used to compare two values and return a boolean result (true or false).

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| == | Equal to | a == b |
| != | Not equal to | a != b |
| > | Greater than | a > b |
| < | Less than | a < b |
| >= | Greater than or equal to | a >= b |
| <= | Less than or equal to | a <= b |

Example:

java

Copy

int a = 10, b = 5;

System.out.println(a == b); // Output: false

System.out.println(a != b); // Output: true

System.out.println(a > b); // Output: true

System.out.println(a < b); // Output: false

**3. Logical Operators**

Logical operators are used to combine multiple boolean expressions or values and return a boolean result.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| && | Logical AND (true if both operands are true) | a && b |
| ` |  | ` |
| ! | Logical NOT (inverts the boolean value) | !a |

Example:

java

Copy

boolean a = true, b = false;

System.out.println(a && b); // Output: false

System.out.println(a || b); // Output: true

System.out.println(!a); // Output: false

**4. Assignment Operators**

Assignment operators are used to assign values to variables.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| = | Simple assignment | a = 5 |
| += | Addition assignment | a += 5 |
| -= | Subtraction assignment | a -= 5 |
| \*= | Multiplication assignment | a \*= 5 |
| /= | Division assignment | a /= 5 |
| %= | Modulus assignment | a %= 5 |

Example:

java

Copy

int a = 10;

a += 5; // a = a + 5, so a becomes 15

a -= 3; // a = a - 3, so a becomes 12

a \*= 2; // a = a \* 2, so a becomes 24

a /= 4; // a = a / 4, so a becomes 6

**5. Unary Operators**

Unary operators operate on a single operand. They are used to perform operations such as incrementing or decrementing the value of a variable.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| + | Unary plus (indicates a positive value) | +a |
| - | Unary minus (negates the value) | -a |
| ++ | Increment (increase by 1) | a++ or ++a |
| -- | Decrement (decrease by 1) | a-- or --a |
| ! | Logical NOT (negates boolean value) | !a |

Example:

java

Copy

int a = 5;

System.out.println(++a); // Output: 6 (pre-increment)

System.out.println(a++); // Output: 6 (post-increment)

System.out.println(--a); // Output: 5 (pre-decrement)

System.out.println(a--); // Output: 5 (post-decrement)

**6. Bitwise Operators**

Bitwise operators work on bits and perform bit-by-bit operations.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| & | Bitwise AND | a & b |
| ` | ` | Bitwise OR |
| ^ | Bitwise XOR | a ^ b |
| ~ | Bitwise NOT | ~a |
| << | Left shift | a << b |
| >> | Right shift | a >> b |
| >>> | Unsigned right shift | a >>> b |

Example:

java

Copy

int a = 5; // (in binary: 0101)

int b = 3; // (in binary: 0011)

System.out.println(a & b); // Output: 1 (binary: 0001)

System.out.println(a | b); // Output: 7 (binary: 0111)

System.out.println(a ^ b); // Output: 6 (binary: 0110)

**7. Ternary Operator**

The ternary operator is a shorthand for the if-else statement. It takes three operands and returns one of two values based on a condition.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| ?: | Ternary (conditional) operator | condition ? expr1 : expr2 |

Example:

java

Copy

int a = 5, b = 10;

int result = (a > b) ? a : b; // Output: 10

**8. Instance of Operator**

The instanceof operator is used to check whether an object is an instance of a specific class or subclass.

Example:

java

Copy

String str = "Hello";

boolean isString = str instanceof String; // Output: true

**Conclusion:**

Java provides a wide range of operators that you can use to perform various operations, from basic arithmetic to more complex bitwise manipulations and logical comparisons. Understanding how and when to use these operators is essential for writing efficient Java programs.