Java Operators

1. Arithmetic Operators

- **Description:** Perform basic mathematical operations.
- **Operators:** `+`, `-`, `*`, `/`, `%`
- **Example Code:**

```java
public class ArithmeticExample {
    public static void main(String[] args) {
        int a = 10;
        int b = 3;

        System.out.println("Addition: " + (a + b));         // Output: 13
        System.out.println("Subtraction: " + (a - b));      // Output: 7
        System.out.println("Multiplication: " + (a * b));   // Output: 30
        System.out.println("Division: " + (a / b));         // Output: 3
        System.out.println("Modulus: " + (a % b));          // Output: 1
    }
}
```

---

2. Comparison Operators

- **Description:** Compare two values; result is a boolean (`true` or `false`).
- **Operators:** `==`, `!=`, `>`, `<`, `>=`, `<=`
- **Example Code:**

```java
public class ComparisonExample {
    public static void main(String[] args) {
        int a = 10;
        int b = 3;

        System.out.println("Equal to: " + (a == b));        // Output: false
        System.out.println("Not equal to: " + (a != b));    // Output: true
        System.out.println("Greater than: " + (a > b));     // Output: true
        System.out.println("Less than: " + (a < b));        // Output: false
        System.out.println("Greater than or equal: " + (a >= b));  // Output: true
        System.out.println("Less than or equal: " + (a <= b));     // Output: false
    }
}
```

---

3. Logical Operators

- **Description:** Perform logical operations on boolean values.
- **Operators:** `&&` (AND), `||` (OR), `!` (NOT)
- **Example Code:**

```java
public class LogicalExample {
    public static void main(String[] args) {
        boolean x = true;
        boolean y = false;

        System.out.println("AND: " + (x && y));             // Output: false
        System.out.println("OR: " + (x || y));              // Output: true
        System.out.println("NOT: " + (!x));                 // Output: false
    }
}
```

---

4. Bitwise Operators

- **Description:** Operate on individual bits of integer types.
- **Operators:** `&` (AND), `|` (OR), `^` (XOR), `~` (NOT), `<<` (left shift), `>>` (right shift), `>>>` (unsigned right shift)
- **Example Code:**

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

```
        int a = 5;    // 0101 in binary
        int b = 3;    // 0011 in binary

        System.out.println("Bitwise AND: " + (a & b));      // Output: 1
        System.out.println("Bitwise OR: " + (a | b));       // Output: 7
        System.out.println("Bitwise XOR: " + (a ^ b));      // Output: 6
        System.out.println("Bitwise NOT: " + (~a));         // Output: -6
    }
}
```

---

5. Unary Operators

- **Description:** Operate on a single operand.
- **Operators:** `+`, `-`, `++`, `--`, `!`
- **Example Code:**

```
public class UnaryExample {
    public static void main(String[] args) {
        int a = 5;
        int b = -a;

        System.out.println("Unary Minus: " + b);             // Output: -5
        System.out.println("Post-Increment: " + (a++));      // Output: 5, then a becomes 6
        System.out.println("Pre-Increment: " + (++a));       // Output: 7
        System.out.println("Post-Decrement: " + (a--));      // Output: 7, then a becomes 6
        System.out.println("Pre-Decrement: " + (--a));       // Output: 5
    }
}
```

---

6. Ternary Operator

- **Description:** A shorthand for `if-else` statements.
- **Syntax:** `condition ? value_if_true : value_if_false`
- **Example Code:**

```
public class TernaryExample {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        int max = (a > b) ? a : b;

        System.out.println("The maximum value is: " + max); // Output: 20
    }
}
```

---

7. Assignment Operators

- **Description:** Assign values to variables.
- **Operators:** `=`, `+=`, `-=`, `*=`, `/=`, `%=`
- **Example Code:**

```
public class AssignmentExample {
    public static void main(String[] args) {
        int a = 10;
        a += 5;    // Same as a = a + 5
        System.out.println("After +=: " + a);              // Output: 15

        a -= 3;    // Same as a = a - 3
        System.out.println("After -=: " + a);              // Output: 12

        a *= 2;    // Same as a = a * 2
        System.out.println("After *=: " + a);              // Output: 24

        a /= 4;    // Same as a = a / 4
        System.out.println("After /=: " + a);              // Output: 6
    }
}
```

---

8. Other Operators

- **Instanceof Operator:** Checks whether an object is an instance of a specific class or subclass.
- **Dot (`.`) Operator:** Accesses members (fields and methods) of a class or an object.