**Lecture on Java Operators**

**1. Introduction to Java Operators**

In Java, operators are special symbols or keywords used to perform operations on variables and values. They are the building blocks of Java programming, enabling you to manipulate data and control program flow.

**2. Types of Java Operators**

Java provides several categories of operators:

1. **Arithmetic Operators**
2. **Relational (Comparison) Operators**
3. **Logical Operators**
4. **Bitwise Operators**
5. **Assignment Operators**
6. **Unary Operators**
7. **Ternary Operator**

**3. Categories of Operators**

**a. Arithmetic Operators**

These operators perform mathematical operations.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| + | Addition | a + b | Sum of a and b |
| - | Subtraction | a - b | Difference of a and b |
| \* | Multiplication | a \* b | Product of a and b |
| / | Division | a / b | Quotient of a and b |
| % | Modulus (Remainder) | a % b | Remainder when a is divided by b |

* **Example**:

java

int a = 10, b = 3;

System.out.println("Addition: " + (a + b)); // 13

System.out.println("Modulus: " + (a % b)); // 1

**b. Relational (Comparison) Operators**

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

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| == | Equal to | a == b | true if a is equal to b |
| != | Not equal to | a != b | true if a is not equal to b |
| > | Greater than | a > b | true if a is greater than b |
| < | Less than | a < b | true if a is less than b |
| >= | Greater than or equal to | a >= b | true if a is greater than or equal to b |
| <= | Less than or equal to | a <= b | true if a is less than or equal to b |

* **Example**:

java

int a = 5, b = 10;

System.out.println(a > b); // false

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

**c. Logical Operators**

Used for combining boolean expressions.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| && | Logical AND | a > 5 && b < 10 | true if both conditions are true |
| || | Logical OR | ` | Logical OR |
| ! | Logical NOT (negation) | !(a > b) | true if a is not greater than b |

* **Example**:

java

int a = 10, b = 5;

System.out.println(a > b && b > 0); // true

System.out.println(!(a < b)); // true

**e. Assignment Operators**

Used to assign values to variables.

| **Operator** | **Description** | **Example** | **Equivalent to** |
| --- | --- | --- | --- |
| = | Assign value | a = b | Assign b to a |
| += | Add and assign | a += b | a = a + b |
| -= | Subtract and assign | a -= b | a = a - b |
| \*= | Multiply and assign | a \*= b | a = a \* b |
| /= | Divide and assign | a /= b | a = a / b |
| %= | Modulus and assign | a %= b | a = a % b |

* **Example**:

java

int a = 10;

a += 5; // a = a + 5

System.out.println(a); // 15

**f. Unary Operators**

Operate on a single operand.

| **Operator** | **Description** | **Example** | **Result** |
| --- | --- | --- | --- |
| + | Unary plus (no effect) | +a | Positive value of a |
| - | Unary minus | -a | Negative value of a |
| ++ | Increment (pre/post) | ++a or a++ | Increases value by 1 |
| -- | Decrement (pre/post) | --a or a-- | Decreases value by 1 |
| ! | Logical NOT | !a | Inverts boolean value |

* **Example**:

java

int a = 5;

System.out.println(++a); // 6

System.out.println(a--); // 6

System.out.println(a); // 5

**g. Ternary Operator**

A shorthand for if-else statements.  
**Syntax**:

java

variable = (condition) ? valueIfTrue : valueIfFalse;

* **Example**:

java

int a = 10, b = 20;

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

System.out.println("Max: " + max); // 20

**4. Best Practices for Using Operators**

1. Use meaningful variable names for better readability.
2. Be cautious with integer division (/), as it discards the decimal part.
3. Avoid overusing increment (++) and decrement (--) in complex expressions.
4. Always parenthesize conditions when using logical operators to avoid ambiguity.

**5. Homework/Practice**

1. Write a program demonstrating the use of all arithmetic operators.
2. Write a program using the ternary operator to find the maximum of three numbers.
3. Use relational and logical operators in a program to create a simple eligibility checker (e.g., for voting).