**Agenda**

• What is an operator?

• Different type of operators in java?

• Perform the operations using the operators

In Java, an operator is a special symbol or keyword used to perform operations on variables and values. Operators are essential for performing calculations, comparisons, and manipulating data.

**Type of Operators:**

* Arithmetic Operators
* Unary (or) Increment/Decrement Operators
* Relational/Comparison Operators
* Conditional/Logical Operators
* Assignment Operators
* Ternary Operator

**Arithmetic Operators:**

Used to perform mathematical operations like addition, subtraction, multiplication, etc.

+ (Addition)

- (Subtraction)

\* (Multiplication)

/ (Division)

% (Modulus)

**Example:**

|  |
| --- |
| public class ArithmeticExample  {  public static void main(String[] args)  {  int a = 20, b = 10;  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("Modulus: " + (a % b));  }  } |

**Unary Operators:**

Operate on a single operand.

++ (Increment)

-- (Decrement)

|  |
| --- |
| public class UnaryExample  {  public static void main(String[] args)  {  int a = 10;  // Prints 10, then increments  System.out.println("Post-increment: " + (a++));  // Prints 11  System.out.println("After increment: " + a);  // Increments first, then prints  System.out.println("Pre-increment: " + (++a));  }  } |

**Relational/Comparison Operators:**

Used to compare values

== (equal to)

!= (not equal to)

> (greater than)

< (less than)

>= (greater than or equal to)

<= (less than or equal to)

**Example:**

|  |
| --- |
| public class RelationalExample  {  public static void main(String[] args)  {  int x = 30, y = 20;  System.out.println("x == y: " + (x == y));  System.out.println("x != y: " + (x != y));  System.out.println("x > y: " + (x > y));  System.out.println("x < y: " + (x < y));  System.out.println("x >= y: " + (x >= y));  System.out.println("x <= y: " + (x <= y));  }  } |

**Conditional Operators/ Logical Operators:**

Used to perform logical operations on Boolean values.

**Operators:** && (AND), || (OR), ! (NOT)

**&& (Conditional AND)**

true - true 🡪 true

true - false 🡪 false

false - true 🡪 false

false - false 🡪 false

**|| (Conditional OR)**

true || true 🡪 true

true || false 🡪 true

false || true 🡪 true

false || false 🡪 false

**! (Conditional NOT)**

The **Logical NOT** operator inverts the boolean value.

If the value is true, it becomes false, and vice versa.

**Example:**

|  |
| --- |
| public class LogicalExample  {  public static void main(String[] args)  {  boolean a = true, b = false;  System.out.println("a && b: " + (a && b));  System.out.println("a || b: " + (a || b));  System.out.println("!a: " + (!a));  }  } |

**Assignment Operators:**

Used to assign values to variables

Operators: =, +=, -=, \*=, /=, %=

**Example:**

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

**Ternary Operator (? :)**

A shorthand for **if-else**.

**Syntax:** condition ? value\_if\_true : value\_if\_false;

**Example:**

|  |
| --- |
| int a = 10, b = 20;  int max = (a > b) ? a : b;  System.out.println("Maximum is: " + max); |

If a > b is true, max will be assigned a, otherwise b.