OPERATORS

- operators are special symbols or keywords used to perform operations on variables and values.
- They are a core part of all programming languages and allow developers to perform arithmetic, comparison, assignment, logical, and other operations.
- Below is a detailed overview of the different types of operators, along with examples for each category, including the ternary operator.

1. Arithmetic Operators

- Used to perform basic mathematical operations such as addition, subtraction, multiplication, and division.

Operator	Description	Example
+	Addition	a + b (Adds a and b)
-	Subtraction	a - b (Subtracts b from a)
*	Multiplication	a * b (Multiplies a and b)
/	Division	a / b (Divides a by b)
%	Modulus	a % b (Remainder of a divided by b) |

2. Relational (Comparison) Operators

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

Operator	Description	Example
==	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)

3. Logical Operators

Logical operators are used to combine conditional statements.

Operator	Description	Example
&&	Logical AND	(a > b) && (a < c) (True if both conditions are true)
||	Logical OR	(a < c) (True if at least one condition is true)
!	Logical NOT	!(a > b (True if `a > b is false)

4. Assignment Operators

Used to assign values to variables.

Operator	Description	Example
==	Assign	a = 5 (Assigns 5 to a)
+=	Add and assign	a += 5 (Same as a = a + 5)
-=	Subtract and assign	a -= 5 (Same as a = a - 5)
*=	Multiply and assign	a *= 5 (Same as a = a * 5)
/=	Divide and assign	a /= 5 (Same as a = a / 5)
%=	Modulus and assign	a %= 5 (Same as a = a % 5`)

5. Bitwise Operators

These operators perform operations on binary numbers.

Operator	Description	Example
&	AND	a & b (Bitwise AND)
|	OR	a | b (Bitwise OR)
^	XOR	a ^ b (Bitwise XOR)

~	NOT	~a (Bitwise NOT)
<<	Left shift	a << 2 (Shifts a left by 2 bits)
>>	Right shift	a >> 2 (Shifts a right by 2 bits)

6. Unary Operators

These operate on a single operand.

Operator	Description	Example
+	Unary plus	+a (Returns the value of a)
-	Unary minus	-a (Negates the value of a)
++	Increment	a++ or ++a (Increases a by 1)
--	Decrement	a-- or --a (Decreases a by 1)

7. Ternary (Conditional) Operator

The ternary operator is a shorthand for an `if-else` statement. It takes three operands.

Syntax:

Condition? expression_if_true : expression_if_false;

Example:

java

int a = 10, b = 20;

int max = (a > b) ? a: b; // If `a` is greater, `max` gets the value of `a, otherwise `b`.

System.out.println(max);   // Output: 20

Here, if the condition `(a > b)` is true, `max` will take the value of `a`. If false, it will take the value of `b`.

8. Special Operators

Some languages offer unique operators for specific purposes.

Operator	Description	Example
Instance of String	Checks if an object is an instance of a class	Object instance of String
?	Ternary operator	`(a > b) ? a : b
->	Lambda expression(Java)	`(x) -> x * 2
::	Method references(Java)	System.out::println