DAY 8

1. What is an Operator?

An **operator** is a special symbol that performs an operation on one or more values. These values are called **operands**. For example, in the expression 5 + 3, the + is the operator and 5 and 3 are the operands. Operators are used to perform mathematical calculations, compare values, and control the flow of a program.

Real-time use case:

Think about an online shopping cart. When you add an item, an operator is used to calculate the new total. If a coupon is applied, another operator subtracts the discount from the subtotal.

---

2. Different Types of Operators in Java

Java has a rich set of operators, which can be categorized as follows:

a) Arithmetic Operators

These are used to perform basic mathematical operations.

Operator	Description	Example
+	Addition	int sum = 10 + 5; // sum is 15
-	Subtraction	int diff = 10 - 5; // diff is 5
*	Multiplication	int prod = 10 * 5; // prod is 50
/	Division	int div = 10 / 5; // div is 2
%	Modulus (Remainder)	int rem = 10 % 3; // rem is 1

b) Relational (Comparison) Operators

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

Operator	Description	Example
==	Equal to	boolean result = (5 == 5); // result is true
!=	Not equal to	boolean result = (5 != 3); // result is true
>	Greater than	boolean result = (5 > 3); // result is true

Operator	Description	Example
<	Less than	boolean result = (5 < 3); // result is false
>=	Greater than or equal to	boolean result = (5 >= 5); // result is true
<=	Less than or equal to	boolean result = (5 <= 3); // result is false

c) Logical Operators

These are used to combine multiple conditional statements and return a boolean result.

Operator	Description	Example
&&	Logical AND	if (age > 18 && hasLicense)
`		`
!	Logical NOT	if (!isLoggedIn)

d) Assignment Operators

These are used to assign values to variables. The most common is the = operator. Compound assignment operators (+=, -=, etc.) are shortcuts.

Operator	Description	Example
=	Assigns a value	int x = 10;
+=	Adds and assigns	x += 5; // x is now 15
-=	Subtracts and assigns	x -= 2; // x is now 13
*=	Multiplies and assigns	x *= 3; // x is now 39

DAY 8

3. Performing Operations with Operators

Let's see how these operators work together in a simple Java program.

Java

```
public class OperatorExample {
    public static void main(String[] args) {
        int a = 15;
        int b = 4;

        // Arithmetic Operators
        System.out.println("a + b = " + (a + b)); // Output: a + b = 19
        System.out.println("a % b = " + (a % b)); // Output: a % b = 3

        // Relational Operators
        System.out.println("a > b is " + (a > b)); // Output: a > b is true

        // Logical Operators
        boolean isTall = true;
        boolean isYoung = false;
        System.out.println("Tall AND Young: " + (isTall && isYoung)); // Output: Tall AND Young: false

        // Assignment Operator
        int c = 20;
        c /= 5;
        System.out.println("c is now: " + c); // Output: c is now: 4
    }
}
```

DAY 8

4. Interview Questions on Operators

1. **What is the difference between the == and equals() operator in Java?**
    - **Expected Answer:** == is a relational operator that compares memory addresses for objects (or values for primitives). The equals() method, on the other hand, is used to compare the content of two objects.
2. **Explain the modulus operator (%) and provide a real-time example.**
    - **Expected Answer:** The modulus operator returns the remainder of a division. A real-time example is checking if a number is even or odd (if number % 2 == 0, it's even).
3. **What is the purpose of the logical && and || operators?**
    - **Expected Answer:** && (AND) returns true only if both conditions are true. || (OR) returns true if at least one of the conditions is true.
4. **Can you explain the difference between a = a + 1 and a++?**
    - **Expected Answer:** a = a + 1 is an arithmetic assignment. a++ is a unary operator called the increment operator, which is a shorthand for adding 1.
5. **What is operator precedence, and why is it important?**
    - **Expected Answer:** Operator precedence is the order in which operators are evaluated in an expression. It's important because it determines the final result of a complex expression (e.g., * and / have higher precedence than + and -).