Day 7 – Methods and Parameter Passing in Java

Objective:

To learn how to define and call methods in Java, understand parameter passing mechanisms, and use methods to organize and reuse code efficiently.

Content:

Today, I studied **methods** in Java, which allow breaking a program into smaller reusable blocks. Methods improve readability, reusability, and modular design.

1. Defining and Calling Methods

A method is a group of statements that performs a specific task.
It can return a value or be `void` (no return value).

Syntax:

```
returnType methodName(parameters) {
        return value;
}
```

Example:

```
class Calculator {
    int add(int a, int b) {
        return a + b;
    }
    public static void main(String[] args) {
        Calculator c = new Calculator();
        System.out.println("Sum: " + c.add(5, 10));
    }
}
```

---

2. Types of Methods

- **Predefined Methods:** Already available in Java (e.g., `Math.sqrt()`, `System.out.println()`).
- **User-Defined Methods:** Created by the programmer for specific tasks.

---

3. Parameter Passing in Java

Java supports **pass-by-value**, meaning a copy of the actual parameter is passed to the method.

Example:

```
void display(String name) {
    System.out.println("Hello, " + name);
}
```

Method Overloading:
Multiple methods can share the same name but differ in parameter type or count.

```
int multiply(int a, int b) { return a * b; }
double multiply(double a, double b) { return a * b; }
```

---

4. Return Statement

Used to return a value from a method back to the caller.

```
int square(int n) {
    return n * n;
}
```

Learning Outcome:

Learned how to define, call, and organize methods in Java programs.
Understood the concept of parameter passing and method overloading.
Gained practical experience in creating reusable and modular code using methods.