**Internship Report – Day 8**

Date: 11 June 2025

Topic: Object-Oriented Programming (OOP) Concepts in Java

Summary:

On the eighth day of our internship, we explored the core concepts of Object-Oriented Programming (OOP) in Java. These concepts form the foundation of Java and are crucial for designing clean, reusable, and maintainable code.

We covered the following key principles:

1. Class and Object:

A class is a blueprint for creating objects.

Objects are instances of classes that contain state (fields) and behavior (methods).

Example:

class Car {

String color;

void drive() {

System.out.println("Driving the car");

}

}

2. Encapsulation:

Binding data and methods together while keeping the details hidden using private access modifiers.

Achieved using getter and setter methods.

Example:

class Student {

private int age;

public void setAge(int a) { age = a; }

public int getAge() { return age; }

}

3. Inheritance:

Enables a class to inherit properties and behaviors from another class using the extends keyword.

Promotes code reusability.

Example:

class Animal {

void sound() { System.out.println("Animal sound"); }

}

class Dog extends Animal {

void bark() { System.out.println("Bark"); }

}

4. Polymorphism:

Allows one name to have many forms—method overloading and overriding.

Enhances flexibility and scalability of code.

Example:

class Shape {

void draw() { System.out.println("Drawing shape"); }

}

class Circle extends Shape {

void draw() { System.out.println("Drawing circle"); }

}

5. Abstraction:

Hiding complex implementation details and showing only the necessary features.

Achieved using abstract classes and interfaces.

Example:

abstract class Animal {

abstract void makeSound();

}

Learning Outcome:

Understood the four main pillars of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Learned how these principles are implemented in Java using classes, methods, and interfaces.

Practiced creating simple Java programs applying OOP concepts.

Conclusion:

Today’s session provided a solid understanding of object-oriented programming in Java, which is

principles will help in writing scalable and maintainable software. This knowledge forms the basis for advanced Java topics and frameworks.