Day 9 – Object-Oriented Programming Concepts – Part 2

Objective:

To understand advanced Object-Oriented Programming (OOP) concepts in Java such as inheritance, polymorphism, abstraction, and encapsulation.

Content:

Today, I learned about the **core principles of OOP** that make Java a powerful, modular, and reusable programming language. These principles help in building real-world applications with well-organized code.

---

1. Inheritance

Inheritance allows one class to acquire the properties and methods of another class, promoting code reuse.

Syntax:

```
class Parent {
    void display() {
        System.out.println("This is the parent class");
    }
}
class Child extends Parent {
    void show() {
        System.out.println("This is the child class");
    }}
public class Main {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.display();
        obj.show();
    }}
```

Output:

```
This is the parent class
This is the child class
```

---

2. Polymorphism

Polymorphism means *"many forms"*. It allows one interface to be used for different types of actions.
There are two types:

- **Compile-time polymorphism** (method overloading)
- **Runtime polymorphism** (method overriding)

Example:

```
class Shape {
    void draw() { System.out.println("Drawing a shape"); }
}
class Circle extends Shape {
    void draw() { System.out.println("Drawing a circle"); }
}
```

---

3. Abstraction

Abstraction hides unnecessary details and shows only essential features.
It is implemented using **abstract classes** or **interfaces**.

```
abstract class Animal {
    abstract void sound();
}
class Dog extends Animal {
    void sound() { System.out.println("Barks"); }
}
```

---

4. Encapsulation

Encapsulation binds data and methods together and protects them from unauthorized access using **private** variables and **getter/setter methods**.

Example:

```
class Account {
    private int balance = 1000;
    public int getBalance() { return balance; }
    public void setBalance(int amt) { balance = amt; }
}
```

Learning Outcome:

Understood and implemented OOP concepts like inheritance, polymorphism, abstraction, and encapsulation.
Learned how these concepts improve modularity, reusability, and security in Java programs.