Java Code Implementation

// Abstract class for Abstraction

abstract class Person {

private String name;

private int age;

// Default Constructor

Person() {

System.out.println("Default Person created");

}

// Parameterized Constructor

Person(String name, int age) {

this.name = name;

this.age = age;

}

// Copy Constructor

Person(Person p) {

this.name = p.name;

this.age = p.age;

}

// Encapsulation - Getters & Setters

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public int getAge() { return age; }

public void setAge(int age) { this.age = age; }

// Abstract method (to be implemented in child classes)

abstract void displayInfo();

}

// Child Class 1 - Student (Inheritance)

class Student extends Person {

private int rollNo;

private double marks;

// Default Constructor

Student() {

super(); // calls Person default

System.out.println("Default Student created");

}

// Parameterized Constructor

Student(String name, int age, int rollNo, double marks) {

super(name, age); // calling parent constructor

this.rollNo = rollNo;

this.marks = marks;

}

// Copy Constructor

Student(Student s) {

super(s); // copy Person attributes

this.rollNo = s.rollNo;

this.marks = s.marks;

}

// Encapsulation

public int getRollNo() { return rollNo; }

public void setRollNo(int rollNo) { this.rollNo = rollNo; }

public double getMarks() { return marks; }

public void setMarks(double marks) { this.marks = marks; }

// Polymorphism (method overriding)

@Override

void displayInfo() {

System.out.println("🎓 Student Details:");

System.out.println("Name: " + getName());

System.out.println("Age: " + getAge());

System.out.println("Roll No: " + rollNo);

System.out.println("Marks: " + marks);

System.out.println("-----------------------------");

}

}

// Child Class 2 - Teacher (Inheritance)

class Teacher extends Person {

private String subject;

// Parameterized Constructor

Teacher(String name, int age, String subject) {

super(name, age);

this.subject = subject;

}

// Polymorphism (method overriding)

@Override

void displayInfo() {

System.out.println("👩‍🏫 Teacher Details:");

System.out.println("Name: " + getName());

System.out.println("Age: " + getAge());

System.out.println("Subject: " + subject);

System.out.println("-----------------------------");

}

}

// Main Class

public class Main {

public static void main(String[] args) {

// Using Parameterized Constructor

Student s1 = new Student("Aditya", 20, 101, 88.5);

s1.displayInfo();

// Using Copy Constructor

Student s2 = new Student(s1);

s2.setRollNo(102);

s2.setMarks(91.2);

s2.displayInfo();

// Teacher Object

Teacher t1 = new Teacher("Mrs. Sharma", 35, "Computer Science");

t1.displayInfo();

// Demonstrate Polymorphism

Person p1 = new Student("Riya", 19, 103, 95.0);

Person p2 = new Teacher("Mr. Verma", 40, "Mathematics");

// Dynamic Method Dispatch

p1.displayInfo();

p2.displayInfo();

}

}

🧾 Expected Output

Default Person created

Default Student created

🎓 Student Details:

Name: Aditya

Age: 20

Roll No: 101

Marks: 88.5

-----------------------------

🎓 Student Details:

Name: Aditya

Age: 20

Roll No: 102

Marks: 91.2

-----------------------------

👩‍🏫 Teacher Details:

Name: Mrs. Sharma

Age: 35

Subject: Computer Science

-----------------------------

🎓 Student Details:

Name: Riya

Age: 19

Roll No: 103

Marks: 95.0

-----------------------------

👩‍🏫 Teacher Details:

Name: Mr. Verma

Age: 40

Subject: Mathematics

-----------------------------