**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 10: Implementing the MVC Pattern**

**Scenario:**

You are developing a simple web application for managing student records using the MVC pattern.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named MVCPatternExample.
2. **Define Model Class:**
   * Create a class Student with attributes like name, id, and grade.
3. **Define View Class:**
   * Create a class StudentView with a method displayStudentDetails().
4. **Define Controller Class:**
   * Create a class StudentController that handles the communication between the model and the view.
5. **Test the MVC Implementation:**
   * Create a main class to demonstrate creating a Student, updating its details using StudentController, and displaying them using StudentView.

**DESCRIPTION:**

This program demonstrates the MVC Pattern for a student management system:

* Student is the Model, holding student data.
* StudentView is the View, responsible for displaying data.
* StudentController is the Controller, managing updates between model and view.
* The main class shows how to create and update a student while keeping data and UI separate.

**PROGRAM:**

public class MVCPatternExample {

public static void main(String[] args) {

Student model = new Student("Keerthi", "SCT123", "A");

StudentView view = new StudentView();

StudentController controller = new StudentController(model, view);

controller.updateView();

controller.setStudentName("Riya");

controller.setStudentGrade("A+");

controller.updateView();

}

}

class Student {

private String name;

private String id;

private String grade;

public Student(String name, String id, String grade) {

this.name = name;

this.id = id;

this.grade = grade;

}

public String getName() {

return name;

}

public String getId() {

return id;

}

public String getGrade() {

return grade;

}

public void setName(String name) {

this.name = name;

}

public void setGrade(String grade) {

this.grade = grade;

}

}

class StudentView {

public void displayStudentDetails(String name, String id, String grade) {

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

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

System.out.println("ID: " + id);

System.out.println("Grade: " + grade);

}

}

class StudentController {

private Student model;

private StudentView view;

public StudentController(Student model, StudentView view) {

this.model = model;

this.view = view;

}

public void setStudentName(String name) {

model.setName(name);

}

public void setStudentGrade(String grade) {

model.setGrade(grade);

}

public void updateView() {

view.displayStudentDetails(model.getName(), model.getId(), model.getGrade());

}

}

**OUTPUT:**

****