**Exercise 10: Implementing the MVC Pattern**

**Step 1: Create a New Java Project**

Create a new Java project named MVCPatternExample using your preferred IDE.

**Step 2: Define Model Class**

Create a class Student with attributes like name, id, and grade, and provide getters and setters.

// Student.java

public class Student {

private String name;

private String id;

private String grade;

// Constructor

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

this.name = name;

this.id = id;

this.grade = grade; }

// Getters and Setters

public String getName() {

return name;}

public void setName(String name) {

this.name = name;}

public String getId() {

return id;}

public void setId(String id) {

this.id = id;}

public String getGrade() {

return grade; }

public void setGrade(String grade) {

this.grade = grade;

}

}

**Step 3: Define View Class**

Create a class StudentView with a method displayStudentDetails() to show the student's details.

// StudentView.java

public class StudentView {

public void displayStudentDetails(String studentName, String studentId, String studentGrade) {

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

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

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

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

}

}

**Step 4: Define Controller Class**

Create a class StudentController that handles the communication between the model and the view.

// StudentController.java

public 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 String getStudentName() {

return model.getName();}

public void setStudentId(String id) {

model.setId(id); }

public String getStudentId() {

return model.getId(); }

public void setStudentGrade(String grade) {

model.setGrade(grade);}

public String getStudentGrade() {

return model.getGrade();

}

public void updateView() {

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

}

}

**Step 5: Test the MVC Implementation**

Create a main class to demonstrate creating a Student, updating its details using StudentController, and displaying them using StudentView.

// MVCPatternExample.java

public class MVCPatternExample {

public static void main(String[] args) {

// Create a Student model

Student student = new Student("John Doe", "S12345", "A");

// Create a StudentView

StudentView view = new StudentView();

// Create a StudentController

StudentController controller = new StudentController(student, view);

// Display the student details

controller.updateView();

// Update student details

controller.setStudentName("Jane Doe");

controller.setStudentId("S54321");

controller.setStudentGrade("B");

// Display the updated student details

controller.updateView();

}

}