**Exercise 10: Implementing the MVC Pattern**

**Scenario:**

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

**StudentController.java :-**

// Source code is decompiled from a .class file using FernFlower decompiler.

package controller;

import model.Student;

import view.StudentView;

public class StudentController {

   private final Student student;

   private final StudentView view;

   public StudentController(Student var1, StudentView var2) {

      this.student = var1;

      this.view = var2;

   }

   public void setStudentName(String var1) {

      this.student.setName(var1);

   }

   public void setStudentGrade(String var1) {

      this.student.setGrade(var1);

   }

   public void updateView() {

      this.view.displayStudentDetails(this.student);

   }

}

**Student.java :-**

package model;

public class Student {

    private String id;

    private String name;

    private String grade;

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

        this.id = id;

        this.name = name;

        this.grade = grade;

    }

    // Getters

    public String getId() { return id; }

    public String getName() { return name; }

    public String getGrade() { return grade; }

    // Setters

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

    public void setGrade(String grade) { this.grade = grade; }

}

**StudentView.java :-**

package view;

import model.Student;

public class StudentView {

    public void displayStudentDetails(Student student) {

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

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

        System.out.println("ID     : " + student.getId());

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

        System.out.println("Grade  : " + student.getGrade());

    }

}

**Main.java :-**

import model.Student;

import view.StudentView;

import controller.StudentController;

public class Main {

    public static void main(String[] args) {

        // Create student and view

        Student student = new Student("101", "Naresh", "A");

        StudentView view = new StudentView();

        // Create controller

        StudentController controller = new StudentController(student, view);

        // Display initial data

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

        controller.updateView();

        // Modify student data

        controller.setStudentName("Naresh Bairaboina");

        controller.setStudentGrade("A+");

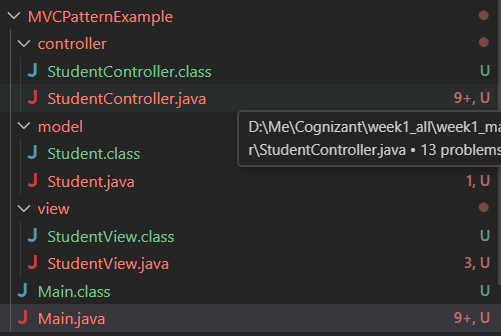
        // Display updated data

        System.out.println("\nAfter Updating:");

        controller.updateView();

    }

}



**Output :-**

