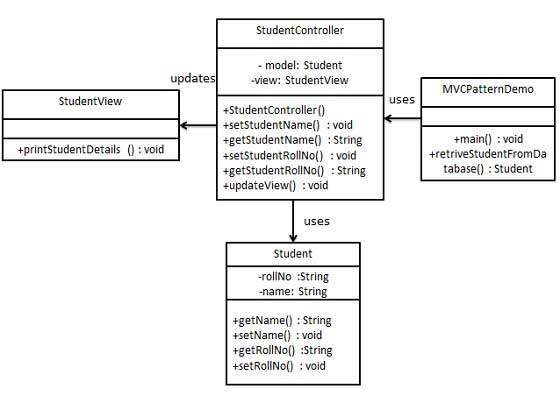
**Exercise Instruction:**

You are requested to complete the installation of the Model-View-Controller (MVC) Pattern. This pattern is used to separate application's concerns.

* **Model** - Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
* **View** - View represents the visualization of the data that model contains.
* **Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

**Task description:**

You are going to create a *Student* object acting as a model.*StudentView* will be a view class which can print student details on console and *StudentController* is the controller class responsible to store data in *Student* object and update view *StudentView* accordingly. *MVCPatternDemo*, our demo class, will use *StudentController* to demonstrate use of MVC pattern.



Figur 1:Class diagram of the MVC model

Step 1

Create Model by installing the class *Student.java*

public class Student { }

Step 2

Create View by implementing the class *StudentView.java.* The class should print the student name and his/ her roll number.

public class StudentView { }

Step 3

Create Controller in *StudentController.java.* The updateView function should call the printStudentDetails function.

public class StudentController {

}

Step 4

After completing Step 1- Step 3, use the *StudentController* method below to demonstrate MVC design pattern usage. *MVCPatternDemo.java*

public class MVCPatternDemo {

public static void main(String[] args) {

//fetch student record based on his roll no from the database

Student model = retriveStudentFromDatabase();

//Create a view : to write student details on console

StudentView view = new StudentView();

StudentController controller = new StudentController(model, view);

controller.updateView();

//update model data

controller.setStudentName("John");

controller.updateView();

}

private static Student retriveStudentFromDatabase(){

Student student = new Student();

student.setName("Robert");

student.setRollNo("10");

return student;

}

}

Step 5

Verify the output. Create the files in your Java Development Environment, compile and compare with our output:

Student:

Name: Robert

Roll No: 10

Student:

Name: John

Roll No: 10