	Course Name: Design Patterns/Thinking LAB	EXPERIMENT NO. 15	
	Course Code: 20CP210P Faculty: Dr. Ketan Sabale	Branch: CSE	Semester: IV
Submitted by: Rhythm Shah **Roll no:** 22BCP071			

Objective: To familiarize students with standard Architectural design patterns.
Experiment: Explain the Model-View-Controller (MVC) design pattern and write a program using any object-oriented programming language to demonstrate the working of MVC design pattern.

Theory:

Model, View, and Controller are the three primary components of an application that are divided into different parts by the MVC design pattern, which simplifies codebase management and maintenance. It also encourages a more modular approach to software development and permits the reusability of components.

1. Model
The Model component in the MVC (Model-View-Controller) design pattern represents the data and business logic of an application. It is responsible for managing the application's data, processing and responding to requests for information from other components, such as the View and the Controller.

2. View
Displays the data from the Model to the user and sends user inputs to the Controller. It receives data from the Model and sends user inputs to the Controller for processing.

3. Controller

Controller acts as an intermediary between the Model and the View. It handles user input and updates the Model accordingly and updates the View to reflect changes in the Model. It contains application logic, such as input validation and data transformation.

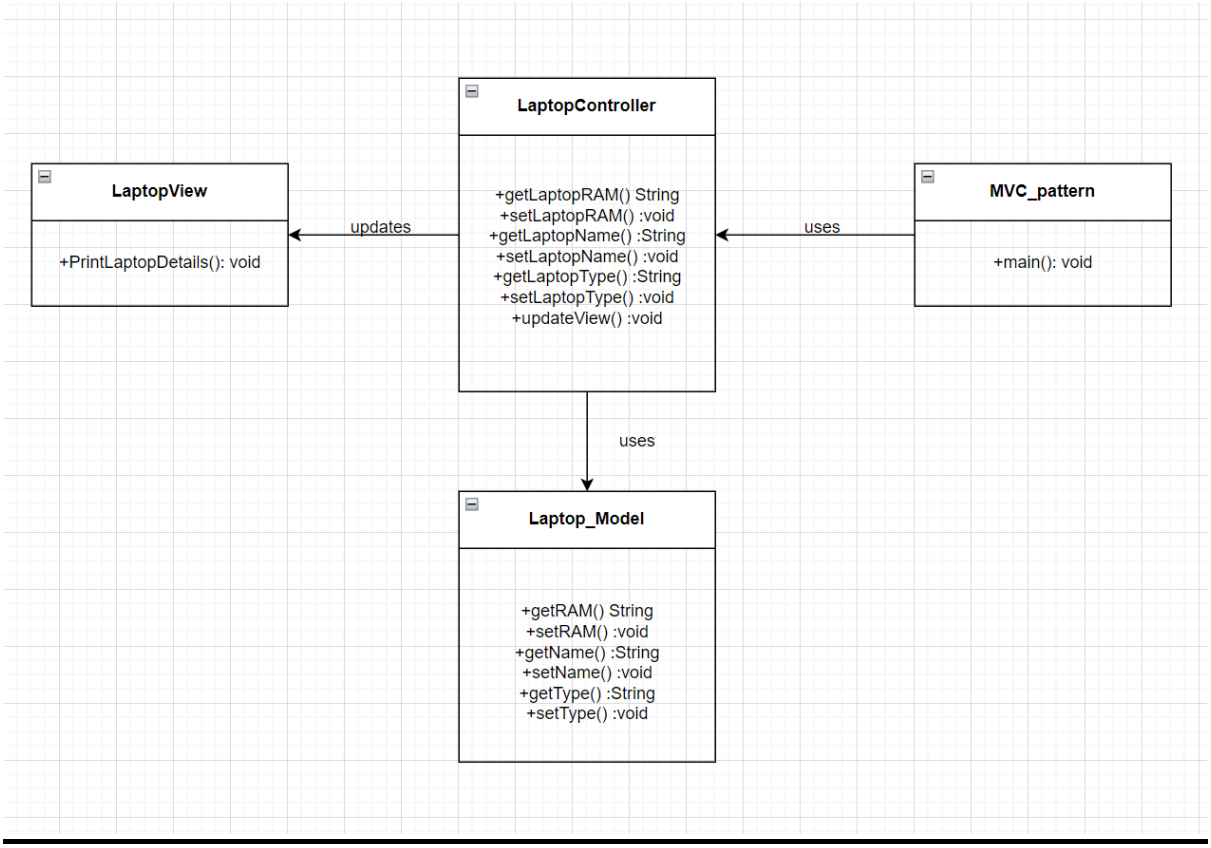
Implementation: -

I have taken an example of Laptop in which the parameters are ram, name and type as they are declared as private therefore getter setter methods are applied. Three classes are made of the main components which are:

1) Laptop Model
2) Laptop View
3) Laptop Controller

And at last, the main method in which objects of each class are created and records are inserted through getter setter methods.

UML Diagram: -



Code: -

```
class Laptop {
    private String ram;
    private String name;
    private String type;

    public String getRAM() {
        return ram;
    }

    public void setRAM(String ram) {
        this.ram = ram;
    }

    public String getName() {
        return name;
    }

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

    public String getType() {
        return type;
    }

    public void setType(String type) {
        this.type = type;
    }
}

class LaptopView {
    public void printLaptopDetails(String laptopName, String laptopRAM, String
laptopType) {
        System.out.println("Laptop:");
        System.out.println("Name: " + laptopName);
        System.out.println("RAM: " + laptopRAM);
        System.out.println("Type: " + laptopType);
    }
}

class LaptopController {
    private Laptop model;
    private LaptopView view;
```

```
    public LaptopController(Laptop model, LaptopView view) {
        this.model = model;
        this.view = view;
    }

    public void setLaptopName(String name) {
        model.setName(name);
    }

    public String getLaptopName() {
        return model.getName();
    }

    public void setLaptopRAM(String ram) {
        model.setRAM(ram);
    }

    public String getLaptopRAM() {
        return model.getRAM();
    }

    public String getLaptopType() {
        return model.getType();
    }

    public void setLaptopType(String type) {
        model.setType(type);
    }


    public void updateView() {
        view.printLaptopDetails(model.getName(), model.getRAM(),
model.getType());
    }
}

public class MVC_pattern {
    public static void main(String[] args) {
        Laptop model = retriveLaptopFromDatabase();

        LaptopView view = new LaptopView();

        LaptopController controller = new LaptopController(model, view);

        controller.updateView();

        controller.setLaptopName("Lenovo");
```

```
        controller.setLaptopRAM("8GB");

        controller.setLaptopType("Office");

        controller.updateView();
    }

    private static Laptop retriveLaptopFromDatabase() {
        Laptop laptop = new Laptop();
        laptop.setName("DELL");
        laptop.setRAM("16GB");
        laptop.setType("Gaming");
        return laptop;
    }
}
```

Output:-

```
Laptop:
Name: DELL
RAM: 16GB
Type: Gaming
Laptop:
Name: Lenovo
RAM: 8GB
Type: Office
PS E:\Fourth sem\Design pattern lab>
```