**Module 1 - Design Patterns and Principles**

**Exercise 9: Implementing the Command Pattern**

**Aim:**

To develope a home automation system where commands can be issued to turn devices on or off using the Command Pattern to achieve this.

**Code:**

interface Command {

    void execute();

}

class Light {

    public void on() {

        System.out.println("Light is ON");

    }

    public void off() {

        System.out.println("Light is OFF");

    }

}

class LightOnCommand implements Command {

    private Light light;

    public LightOnCommand(Light light) {

        this.light = light;

    }

    public void execute() {

        light.on();

    }

}

class LightOffCommand implements Command {

    private Light light;

    public LightOffCommand(Light light) {

        this.light = light;

    }

    public void execute() {

        light.off();

    }

}

class RemoteControl {

    private Command command;

    public void setCommand(Command command) {

        this.command = command;

    }

    public void pressButton() {

        command.execute();

    }

}

public class CommandPatternExample {

    public static void main(String[] args) {

        Light light = new Light();

        Command onCommand = new LightOnCommand(light);

        Command offCommand = new LightOffCommand(light);

        RemoteControl remote = new RemoteControl();

        remote.setCommand(onCommand);

        remote.pressButton();

        remote.setCommand(offCommand);

        remote.pressButton();

    }

}

**Output:**

