**Exercise 9: Implementing the Command Pattern**

CODE:

public class CommandPatternExample {

interface Command {

void execute();

}

static class Light {

private String location;

public Light(String location) {

this.location = location;

}

public void turnOn() {

System.*out*.println(location + " light is ON");

}

public void turnOff() {

System.*out*.println(location + " light is OFF");

}

}

static class LightOnCommand implements Command {

private Light light;

public LightOnCommand(Light light) {

this.light = light;

}

@Override

public void execute() {

light.turnOn();

}

}

static class LightOffCommand implements Command {

private Light light;

public LightOffCommand(Light light) {

this.light = light;

}

@Override

public void execute() {

light.turnOff();

}

}

static class RemoteControl {

private Command command;

public void setCommand(Command command) {

this.command = command;

}

public void pressButton() {

if (command != null) {

command.execute();

} else {

System.*out*.println("No command set.");

}

}

}

public static void main(String[] args) {

Light livingRoomLight = new Light("Living Room");

Command lightOn = new LightOnCommand(livingRoomLight);

Command lightOff = new LightOffCommand(livingRoomLight);

RemoteControl remote = new RemoteControl();

remote.setCommand(lightOn);

remote.pressButton();

remote.setCommand(lightOff);

remote.pressButton();

}

}

OUTPUT:

