**Exercise 9:**

**Implementing the Command Patterns:**

Command.java:

**package** command;

**public** **interface** command {

**void** execute();

}

Light.java:

**package** command;

**public** **class** Light {

**public** **void** turnon()

{

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

}

**public** **void** turnOff()

{

System.***out***.println("Light is Off");

}

}

LightOnCommand.java:

**package** command;

**public** **class** LightOnCommand **implements** command{

**private** Light light;

**public** LightOnCommand(Light light)

{

**this**.light=light;

}

**public** **void** execute()

{

light.turnon();

}

}

LightOffCommand.java:

**package** command;

**public** **class** LightOffCommand **implements** command{

**private** Light light;

**public** LightOffCommand(Light light)

{

**this**.light=light;

}

**public** **void** execute()

{

light.turnOff();

}

}

RemoteControl.java:

**package** command;

**public** **class** RemoteControl {

**private** command Command;

**public** **void** setcommand(command Command)

{

**this**.Command=Command;

}

**public** **void** pressButton()

{

Command.execute();

}

}

CommandTest.java:

**package** command;

**public** **class** commandTest {

**public** **static** **void** main(String[] args)

{

Light light=**new** Light();

command lighton=**new** LightOnCommand(light);

command lightoff=**new** LightOffCommand(light);

RemoteControl remote=**new** RemoteControl();

remote.setcommand(lighton);;

remote.pressButton();

remote.setcommand(lightoff);

remote.pressButton();

}

}

**Output:**

