**Exercise 5: Implementing the Decorator Pattern**

**Scenario:**

You are developing a notification system where notifications can be sent via multiple channels (e.g., Email, SMS). Use the Decorator Pattern to add functionalities dynamically.

**File Structure:**

Notifier.java – Interface

EmailNotifier.java – Concrete Component

NotifierDecorator.java – Abstract Decorator

SMSNotifierDecorator.java, SlackNotifierDecorator.java – Concrete Decorators

TestDecorator.java – Main Test Class

**Codes:**

//Notifier.java

public interface Notifier {

void send(String message);

}

//EmailNotifier.java

public class EmailNotifier implements Notifier {

public void send(String message) {

System.out.println("Sending Email: " + message);

}

}

NotifierDecorator.java

public abstract class NotifierDecorator implements Notifier {

protected Notifier wrappedNotifier;

public NotifierDecorator(Notifier notifier) {

this.wrappedNotifier = notifier;

}

public void send(String message) {

wrappedNotifier.send(message);

}

}

SMSNotifierDecorator.java

public class SMSNotifierDecorator extends NotifierDecorator {

public SMSNotifierDecorator(Notifier notifier) {

super(notifier);

}

@Override

public void send(String message) {

super.send(message);

System.out.println("Sending SMS: " + message);

}

}

// SlackNotifierDecorator.java

public class SlackNotifierDecorator extends NotifierDecorator {

public SlackNotifierDecorator(Notifier notifier) {

super(notifier);

}

@Override

public void send(String message) {

super.send(message);

System.out.println("Sending Slack message: " + message);

}

}

//TestDecorator.java

public class TestDecorator {

public static void main(String[] args) {

Notifier emailNotifier = new EmailNotifier();

Notifier smsEmailNotifier = new SMSNotifierDecorator(emailNotifier);

Notifier fullNotifier = new SlackNotifierDecorator(smsEmailNotifier);

System.out.println("Notification via Email, SMS, and Slack:");

fullNotifier.send("Your package has been shipped.");

}

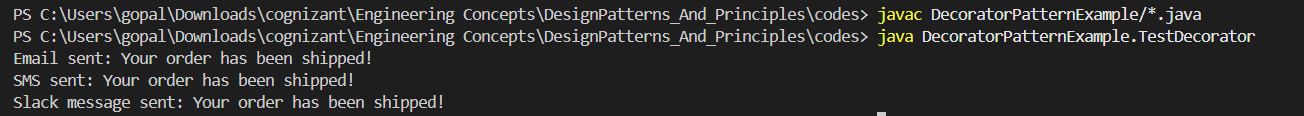
}

**Compile and Run:**

javac BuilderPatternExample/\*.java

java BuilderPatternExample.TestDecorator

**Output:**

****