**WEEK 1**

**Additional Hands-On**

**SOLID PRINCIPLES AND DESIGN PATTERNS:**

**EX:4 Implementing the Adapter Pattern**

The reason to use Adapter design pattern for integrating multiple third-party payment gateways is that,

* It allows incompatible interfaces to work together.
* It provides a standard interface to client code, promoting consistency.
* It supports Open-Closed Principle by allowing new gateways to be added without modifying existing client code.
* It isolates third-party API changes from the rest of the system.

**Target interface, Adaptee and Adapter class (Definition and Implementation):**

public interface PaymentProcessor {

void processPayment(double amount);

}

public class PayPalGateway {

public void makePayment(double amountInDollars) {

System.out.println("Paid $" + amountInDollars + " using PayPal.");

}

}

public class StripeGateway {

public void payAmount(double amt) {

System.out.println("Paid $" + amt + " using Stripe.");

}

}

public class PayPalAdapter implements PaymentProcessor {

private PayPalGateway paypal;

public PayPalAdapter(PayPalGateway paypal) {

this.paypal = paypal;

}

public void processPayment(double amount) {

paypal.makePayment(amount);

}

}

public class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter(StripeGateway stripe) {

this.stripe = stripe;

}

public void processPayment(double amount) {

stripe.payAmount(amount);

}

}

* Each adapter class implements PaymentProcessor (target interface).
* The adapter translates standard method calls into gateway-specific calls.

**Testing the Adapter Implementation using Main class:**

public class Main {

public static void main(String[] args) {

PayPalGateway paypal = new PayPalGateway();

PaymentProcessor paypalProcessor = new PayPalAdapter(paypal);

System.out.println("Using PayPal:");

paypalProcessor.processPayment(250.00);

System.out.println();

StripeGateway stripe = new StripeGateway();

PaymentProcessor stripeProcessor = new StripeAdapter(stripe);

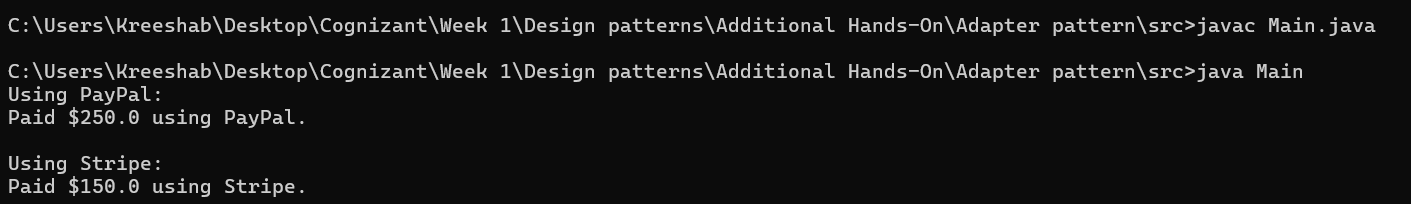
System.out.println("Using Stripe:");

stripeProcessor.processPayment(150.00);

}

}

**TESTING OUTPUT:**

****