**Exercise 8: Strategy Pattern – Payment System**

**Step 1: Strategy Interface – PaymentStrategy.java**

public interface PaymentStrategy {

void pay(double amount);

}

**Step 2: Concrete Strategies**

**// File: CreditCardPayment.java**

public class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

public CreditCardPayment(String cardNumber) {

this.cardNumber = cardNumber;

}

public void pay(double amount) {

System.out.println("Paid ₹" + amount + " using Credit Card: " + cardNumber);

}

}

**// File: PayPalPayment.java**

public class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

public void pay(double amount) {

System.out.println("Paid ₹" + amount + " using PayPal: " + email);

}

}

public class PaymentContext {

private PaymentStrategy strategy;

// Set strategy at runtime

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

// Execute strategy

public void makePayment(double amount) {

if (strategy != null) {

strategy.pay(amount);

} else {

System.out.println("No payment method selected!");

}

}

}

**PaymentContext.java**

public class PaymentContext {

private PaymentStrategy strategy;

// Set strategy at runtime

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

// Execute strategy

public void makePayment(double amount) {

if (strategy != null) {

strategy.pay(amount);

} else {

System.out.println("No payment method selected!");

}

}

}

**TestStrategyPattern.java**

public class TestStrategyPattern {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

// Using Credit Card payment

context.setPaymentStrategy(new CreditCardPayment("1234-5678-9876"));

context.makePayment(2500.00);

// Switching to PayPal payment

context.setPaymentStrategy(new PayPalPayment("anushka@example.com"));

context.makePayment(1500.00);

}

}

**Output:**

