**Exercise 8: Implementing the Strategy Pattern**

**PaymentStrategy.java**

public interface PaymentStrategy {

void pay(double amount);

}

**CreditCardPayment.java**

public class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

public CreditCardPayment(String cardNumber) {

this.cardNumber = cardNumber;

}

*@Override*

public void pay(double amount) {

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

}

}

**PayPalPayment.java**

public class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

*@Override*

public void pay(double amount) {

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

}

}

**PaymentContext.java**

public class PaymentContext {

private PaymentStrategy strategy;

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

public void payAmount(double amount) {

if (strategy == null) {

System.***out***.println("No payment strategy selected.");

} else {

strategy.pay(amount);

}

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

// Pay using Credit Card

context.setPaymentStrategy(new CreditCardPayment("1234-5678-9012-3456"));

context.payAmount(2500);

System.***out***.println("-----------------------------");

// Pay using PayPal

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

context.payAmount(1500);

}

}

Output:

