**Exercise 8: Implementing the Strategy Pattern**

**Code:**

interface PaymentStrategy {

void pay(double amount);

}

class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

public CreditCardPayment(String cardNumber) {

this.cardNumber = cardNumber;

}

@Override

public void pay(double amount) {

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

}

}

class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

@Override

public void pay(double amount) {

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

}

}

class PaymentContext {

private PaymentStrategy paymentStrategy;

public void setPaymentStrategy(PaymentStrategy paymentStrategy) {

this.paymentStrategy = paymentStrategy;

}

public void payAmount(double amount) {

if (paymentStrategy == null) {

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

} else {

paymentStrategy.pay(amount);

}

}

}

public class StrategyPatternExample {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

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

context.payAmount(1800.00);

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

context.payAmount(750.00);

context.setPaymentStrategy(null);

context.payAmount(500.00);

}

}

**Output:**

