**Module 1 - Design Patterns and Principles**

**Exercise 8: Implementing the Strategy Pattern**

**Aim:**

To develope a payment system where different payment methods (e.g., Credit Card, PayPal) can be selected at runtime using the Strategy Pattern to achieve this.

**Code:**

interface PaymentStrategy {

    void pay(double amount);

}

class CreditCardPayment implements PaymentStrategy {

    public void pay(double amount) {

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

    }

}

class PayPalPayment implements PaymentStrategy {

    public void pay(double amount) {

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

    }

}

class PaymentContext {

    private PaymentStrategy strategy;

    public void setStrategy(PaymentStrategy strategy) {

        this.strategy = strategy;

    }

    public void payAmount(double amount) {

        strategy.pay(amount);

    }

}

public class StrategyPatternExample {

    public static void main(String[] args) {

        PaymentContext context = new PaymentContext();

        context.setStrategy(new CreditCardPayment());

        context.payAmount(1500);

        context.setStrategy(new PayPalPayment());

        context.payAmount(2500);

    }

}

Output:

