**NAME : RACHANA J N**

**Superset ID :   6416511**

**TASK 8 : Implementing the Strategy Pattern**

**Scenario:**

You are developing a payment system where different payment methods (e.g., Credit Card, PayPal) can be selected at runtime. Use the Strategy Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **StrategyPatternExample**.
2. **Define Strategy Interface:**
   * Create an interface PaymentStrategy with a method **pay()**.
3. **Implement Concrete Strategies:**
   * Create classes **CreditCardPayment**, **PayPalPayment** that implement **PaymentStrategy**.
4. **Implement Context Class:**
   * Create a class **PaymentContext** that holds a reference to **PaymentStrategy** and a method to execute the strategy.
5. **Test the Strategy Implementation:**
   * Create a test class to demonstrate selecting and using different payment strategies.

**CODE :**

public class TestStrategy {

    public static void main(String[] args) {

        PaymentContext context = new PaymentContext();

        context.setPaymentStrategy(new CreditCardPayment());

        context.executePayment(1200.0);

        context.setPaymentStrategy(new PayPalPayment());

        context.executePayment(850.5);    } }

interface PaymentStrategy {

    void pay(double amount); }

class CreditCardPayment implements PaymentStrategy {

    public void pay(double amount) {

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

    } }

class PayPalPayment implements PaymentStrategy {

    public void pay(double amount) {

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

    } }

class PaymentContext {

    private PaymentStrategy strategy;

    public void setPaymentStrategy(PaymentStrategy strategy) {

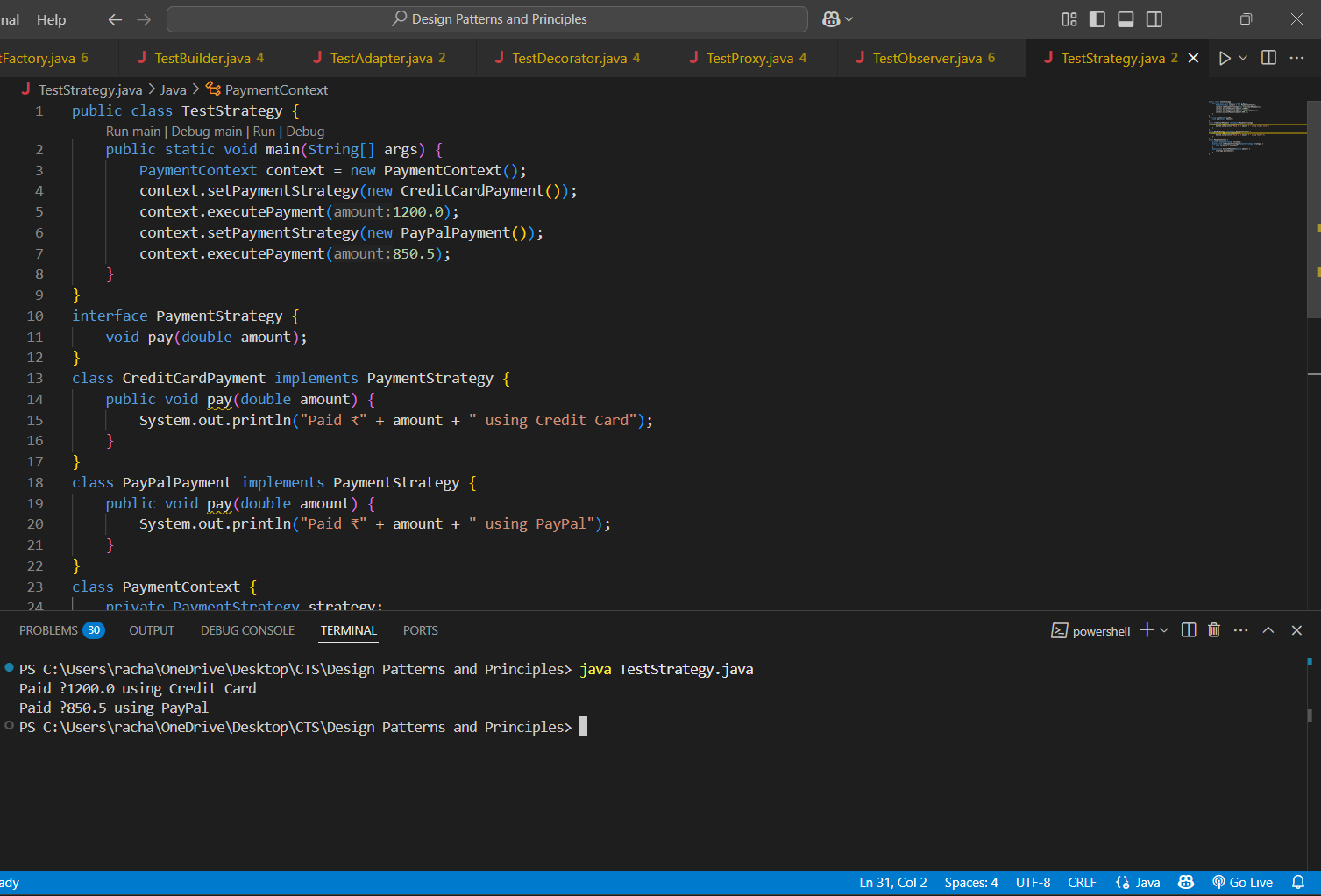
        this.strategy = strategy;   }

    public void executePayment(double amount) {

        strategy.pay(amount);

    } }

**OUTPUT :**

****