**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:**

**PaymentStrategy.java:**

public interface PaymentStrategy {

    void pay(double amount);

}

**PaymentContext.java**

public class PaymentContext {

    private PaymentStrategy paymentstrategy;

    public void setPaymentStrategy(PaymentStrategy paymentStrategy) {

        this.paymentstrategy = paymentStrategy;

    }

    public void pay(double amount) {

        if (paymentstrategy != null) {

            paymentstrategy.pay(amount);

        } else {

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

        }

    }

}

**PayPalPayment,java**

public class PayPalPayment implements PaymentStrategy {

    String email;

    String password;

    PayPalPayment(String email, String password) {

        this.email = email;

        this.password = password;

    }

    public void pay(double amount) {

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

    }

}

**CreditCardPayment.java**

public class CreditCardPayment implements PaymentStrategy {

    String cardnumber;

    String cardHolderNAme;

    String cvv;

    String expiryDate;

    CreditCardPayment(String cardnumber, String cardHolderName, String cvv, String expiryDate) {

        this.cardnumber = cardnumber;

        this.cardHolderNAme = cardHolderName;

        this.cvv = cvv;

        this.expiryDate = expiryDate;

    }

    public void pay(double amount) {

        System.out.println("Paid " + amount + " using credit card");

    }

}

**Main.java**

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = **new** Scanner(System.in);

        PaymentContext paymentContext = **new** PaymentContext();

        try {

            System.out.println("Choose payment method (1: Credit Card, 2: PayPal):");

            int choice = scanner.nextInt();

            scanner.nextLine();

            System.out.println("Enter amount to pay:");

            double amount = scanner.nextDouble();

            scanner.nextLine();

            switch (choice) {

                case 1:

                    System.out.println("Enter card number:");

                    String cardNumber = scanner.nextLine();

                    System.out.println("Enter card holder name:");

                    String cardHolderName = scanner.nextLine();

                    System.out.println("Enter CVV:");

                    String cvv = scanner.nextLine();

                    System.out.println("Enter expiry date (MM/YY):");

                    String expiryDate = scanner.nextLine();

                    PaymentStrategy creditCardPayment = **new** CreditCardPayment(cardNumber, cardHolderName, cvv,

                            expiryDate);

                    paymentContext.setPaymentStrategy(creditCardPayment);

                    break;

                case 2:

                    System.out.println("Enter PayPal email:");

                    String email = scanner.nextLine();

                    System.out.println("Enter PayPal password:");

                    String password = scanner.nextLine();

                    PaymentStrategy payPalPayment = **new** PayPalPayment(email, password);

                    paymentContext.setPaymentStrategy(payPalPayment);

                    break;

                default:

                    System.out.println("Invalid choice");

                    return;

            }

            paymentContext.pay(amount);

        } finally {

            scanner.close();

        }

    }

}