**StratergyPattern Example**

**CreditCardPayment.java**

package com.strategy;

public class CreditCardPayment implements PaymentStrategy {

    private String cardNumber;

    private String cardHolder;

    public CreditCardPayment(String cardNumber, String cardHolder) {

        this.cardNumber = cardNumber;

        this.cardHolder = cardHolder;

    }

    @Override

    public void pay(double amount) {

        System.out.println("Paid $" + amount + " using Credit Card (Holder: " + cardHolder + ")");

    }

}

**Main.java**

package com.strategy;

public class Main {

    public static void main(String[] args) {

        PaymentContext context = new PaymentContext();

        // Use Credit Card Payment

        context.setPaymentStrategy(new CreditCardPayment("1234-5678-9999-0000", "Alice"));

        context.processPayment(250.00);

        System.out.println();

        // Use PayPal Payment

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

        context.processPayment(99.99);

    }

}

**PaymentContext.java**

package com.strategy;

public class PaymentContext {

    private PaymentStrategy strategy;

    public void setPaymentStrategy(PaymentStrategy strategy) {

        this.strategy = strategy;

    }

    public void processPayment(double amount) {

        if (strategy == null) {

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

        } else {

            strategy.pay(amount);

        }

    }

}

PaymentStrategy.java

package com.strategy;

public interface PaymentStrategy {

    void pay(double amount);

}

**PayPalPayment.java**

package com.strategy;

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: " + email + ")");

    }

}

