## ****Exercise 8: Implementing the Strategy Pattern****

### ****Scenario****

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.

### ****Step 1: Create a New Java Project****

Create a new Java project named StrategyPatternExample.

### ****Step 2: Define Strategy Interface****

public interface PaymentStrategy {

void pay(double amount);

}

### ****Step 3: Implement Concrete Strategies****

public class CreditCardPayment implements PaymentStrategy {

@Override

public void pay(double amount) {

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

}

}

public class PayPalPayment implements PaymentStrategy {

@Override

public void pay(double amount) {

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

}

}

### ****Step 4: Implement Context Class****

public class PaymentContext {

private PaymentStrategy strategy;

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

public void pay(double amount) {

if (strategy == null) {

System.out.println("Payment strategy not set!");

} else {

strategy.pay(amount);

}

}

}

### ****Step 5: Test the Strategy Implementation****

public class TestStrategyPattern {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

context.setPaymentStrategy(new CreditCardPayment());

context.pay(250.00);

context.setPaymentStrategy(new PayPalPayment());

context.pay(400.00);

}

}  
  
OUTPUT:

