**Exercise 8: Implementing the Strategy Pattern**

public class StrategyPatternExample {

interface PaymentStrategy {

void pay(double amount);

}

static class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

public CreditCardPayment(String cardNumber) {

this.cardNumber = cardNumber;

}

public void pay(double amount) {

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

}

}

static class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

public void pay(double amount) {

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

}

}

static class PaymentContext {

private PaymentStrategy strategy;

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

public void makePayment(double amount) {

if (strategy == null) {

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

} else {

strategy.pay(amount);

}

}

}

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

System.out.println("Using Credit Card:");

context.setPaymentStrategy(new CreditCardPayment("1234-5678-9012-3456"));

context.makePayment(1500);

System.out.println("\nUsing PayPal:");

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

context.makePayment(2750);

}

}

OUTPUT:

