**Exercise 8: Implementing the Strategy Pattern**

interface PaymentStrategy {

void pay(double amount);

}

class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

private String cardHolderName;

public CreditCardPayment(String cardNumber, String cardHolderName) {

this.cardNumber = cardNumber;

this.cardHolderName = cardHolderName;

}

@Override

public void pay(double amount) {

System.out.println("Paid $" + amount + " using Credit Card [" + cardNumber + "] - Cardholder: " + cardHolderName);

}

}

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 account: " + email);

}

}

class PaymentContext {

private PaymentStrategy paymentStrategy;

public void setPaymentStrategy(PaymentStrategy paymentStrategy) {

this.paymentStrategy = paymentStrategy;

}

public void makePayment(double amount) {

if (paymentStrategy == null) {

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

} else {

paymentStrategy.pay(amount);

}

}

}

public class Main{

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

PaymentStrategy creditCard = new CreditCardPayment("1234-5678-9876-5432", "Alice Johnson");

context.setPaymentStrategy(creditCard);

context.makePayment(250.00);

PaymentStrategy payPal = new PayPalPayment("alice@example.com");

context.setPaymentStrategy(payPal);

context.makePayment(99.99);

}

}

