**Exercise 8: Implementing the Strategy Pattern**

CODE:

public class StrategyPatternExample {

interface PaymentStrategy {

void pay(double amount);

}

static class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

private String name;

public CreditCardPayment(String cardNumber, String name) {

this.cardNumber = cardNumber;

this.name = name;

}

public void pay(double amount) {

System.*out*.println("Paid Rs." + amount + " using Credit Card (Holder: " + name + ")");

}

}

static class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

public void pay(double amount) {

System.*out*.println("Paid Rs." + amount + " via PayPal (Email: " + 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("Please select a payment method.");

} else {

strategy.pay(amount);

}

}

}

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

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

context.makePayment(500.00);

context.setPaymentStrategy(new PayPalPayment("maya99090@gamil.com"));

context.makePayment(250.75);

context.setPaymentStrategy(null);

context.makePayment(100.00);

}

}

OUTPUT:

