**EXERCISE – 8**

**STRATEGY PATTERN EXAMPLE**

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. Card Number: " + cardNumber + ", Card Holder: " + 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. Email: " + email);  
 }  
}  
  
class PaymentContext {  
 private PaymentStrategy paymentStrategy;  
  
 public void setPaymentStrategy(PaymentStrategy paymentStrategy) {  
 this.paymentStrategy = paymentStrategy;  
 }  
  
 public void pay(double amount) {  
 if (paymentStrategy == null) {  
 System.*out*.println("Payment strategy is not set.");  
 } else {  
 paymentStrategy.pay(amount);  
 }  
 }  
}  
  
public class StrategyPatternExample {  
 public static void main(String[] args) {  
 PaymentContext paymentContext = new PaymentContext();  
  
 // Pay using Credit Card  
 PaymentStrategy creditCardPayment = new CreditCardPayment("1234-5678-9876-5432", "Manoj Kumar V");  
 paymentContext.setPaymentStrategy(creditCardPayment);  
 paymentContext.pay(250.00);  
  
 // Pay using PayPal  
 PaymentStrategy payPalPayment = new PayPalPayment("manojkumar@gmail.com");  
 paymentContext.setPaymentStrategy(payPalPayment);  
 paymentContext.pay(100.00);  
 }  
}