**Exercise 8: Implementing the Strategy Pattern**

**Code:**

**PaymentStrategy Interface: -**

public interface PaymentStrategy {  
 void pay(int amount);  
}

**CreditCardPayment Class: -**

public class CreditCardPayment implements PaymentStrategy{  
 private String cardHolderName;  
 private String cardNumber;  
 public CreditCardPayment(String name,String number){  
 this.cardHolderName=name;  
 this.cardNumber=number;  
 }  
 @Override  
 public void pay(int amount) {  
 System.*out*.println(amount+" paid using Credit Card: "+cardNumber+" by "+cardHolderName);  
 }  
}

**PayPalPayment Class: -**

public class PayPalPayment implements PaymentStrategy{  
 private String email;  
 public PayPalPayment(String email){  
 this.email=email;  
 }  
 @Override  
 public void pay(int amount) {  
 System.*out*.println(amount+" paid using PayPal account: "+email);  
 }  
}

**PaymentContext Class: -**

public class PaymentContext {  
 private PaymentStrategy paymentStrategy;  
 public void setPaymentStrategy(PaymentStrategy paymentStrategy){  
 this.paymentStrategy=paymentStrategy;  
 }  
 public void payAmount(int amount){  
 if(paymentStrategy==null){  
 System.*out*.println("No payment strategy selected.");  
 }  
 else{  
 paymentStrategy.pay(amount);  
 }  
 }  
}

**TestClass (Main Class): -**

public class TestClass {  
 public static void main(String[] args){  
 PaymentContext context=new PaymentContext();  
 context.setPaymentStrategy(new CreditCardPayment("Shushma","1234- 5678-9875-2321"));  
 context.payAmount(3000);  
 context.setPaymentStrategy(new

PayPalPayment("shushma.1109@gmail.com"));  
 context.payAmount(5500);  
 }  
}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**