```java
import java.util.Scanner;
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 Rs" + amount + " using Credit Card.");
        System.out.println("Card Holder: " + cardHolderName);
        System.out.println("Card Number: " + maskCardNumber(cardNumber));
    }

    private String maskCardNumber(String cardNumber) {
        if (cardNumber.length() >= 4) {
            return "XXXX-XXXX-XXXX-" + cardNumber.substring(cardNumber.length() - 4);
        } else {
            return "Invalid Card Number";
        }
    }
}
```

```java
29      }
30      class PayPalPayment implements PaymentStrategy {
31          private String email;
32
33          public PayPalPayment(String email) {
34              this.email = email;
35          }
36
37          @Override
38          public void pay(double amount) {
39              System.out.println("Paid Rs " + amount + " using PayPal.");
40              System.out.println("PayPal Email: " + email);
41          }
42      }
43      class PaymentContext {
44          private PaymentStrategy strategy;
45
46          public void setPaymentStrategy(PaymentStrategy strategy) {
47              this.strategy = strategy;
48          }
49
50          public void makePayment(double amount) {
51              if (strategy == null) {
52                  System.out.println(x:"Error: Payment strategy not set!");
53              } else {
54                  strategy.pay(amount);
55              }
56          }
57      }
```

```java
58    public class StrategyPattern {
            Run main | Debug main | Run | Debug
59        public static void main(String[] args) {
60            Scanner scanner = new Scanner(System.in);
61            PaymentContext context = new PaymentContext();
62
63            System.out.println(x:"Enter amount to pay:");
64            double amount = scanner.nextDouble();
65            scanner.nextLine();
66
67            System.out.println(x:"Choose payment method:");
68            System.out.println(x:"1. Credit Card");
69            System.out.println(x:"2. PayPal");
70            int choice = scanner.nextInt();
71            scanner.nextLine();
72
73            switch (choice) {
74                case 1:
75                    System.out.print(s:"Enter card holder name: ");
76                    String name = scanner.nextLine();
77                    System.out.print(s:"Enter card number: ");
78                    String cardNumber = scanner.nextLine();
79                    context.setPaymentStrategy(new CreditCardPayment(cardNumber, name));
80                    break;
81
82                case 2:
83                    System.out.print(s:"Enter PayPal email: ");
84                    String email = scanner.nextLine();
85                    context.setPaymentStrategy(new PayPalPayment(email));
```

J StrategyPattern.java 5 ●

J StrategyPattern.java > ...

```java
58    public class StrategyPattern {
59        public static void main(String[] args) {
82                case 2:
83                    System.out.print(s:"Enter PayPal email: ");
84                    String email = scanner.nextLine();
85                    context.setPaymentStrategy(new PayPalPayment(email));
86                    break;
87
88                default:
89                    System.out.println(x:"Invalid choice. Exiting.");
90                    scanner.close();
91                    return;
92            }
93
94            System.out.println();
95            context.makePayment(amount);
96
97            scanner.close();
98        }
99    }
100
```

EXPLORER

OPEN EDITORS
  ✕ J StrategyPat... 5
COGNIZANT
  J Employee.class
  J EmployeeManageme...
  J EmployeeManageme...
  J FactoryMethodPatter...
  J FinancialForecasting.j...
  J InventoryManageme...
  J LibraryManagementS...
  J ObserverPattern.... 5
  J Product.class
  J ProxyPattern.java 3
  J SingletonPattern.java
  J StrategyPattern.j... 5
  J Task.class
  J TaskList.class
  J TaskManagementSyst...
  J TaskManagementSyst...
  J TaskNode.class

OUTLINE
TIMELINE
PROJECTS
RUN CONFIGURATION
JAVA PROJECTS

J StrategyPattern.java 5 ✕

J StrategyPattern.java > ...
58   public class StrategyPattern {

PROBLEMS 20   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
26780
Choose payment method:
1. Credit Card
2. PayPal
1
Enter card holder name: Ashwini
Enter card number: 2345 9810 5638

Paid Rs26780.0 using Credit Card.
Card Holder: Ashwini
Card Number: XXXX-XXXX-XXXX-5638
PS C:\cognizant> ^C
PS C:\cognizant>
PS C:\cognizant>  c:; cd 'c:\cognizant'; & 'C:\Program Files\Java\jdk-24\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMe
ssages' '-cp' 'C:\Users\sahan\AppData\Roaming\Code\User\workspaceStorage\5c120c78ee1edadaac7ff33613f965a8\redhat.java\jdt_ws\cognizant_43b6c
eb\bin' 'StrategyPattern'
Enter amount to pay:
67000
Choose payment method:
1. Credit Card
2. PayPal
2
Enter PayPal email: harshitha709@gmai.com

Paid Rs 67000.0 using PayPal.
PayPal Email: harshitha709@gmai.com
PS C:\cognizant>
```

Run: StrategyPattern

⊗ 6 ⚠ 14   Indexing completed.   Java: Ready   Ln 1, Col 26   Spaces: 4   UTF-8   CRLF   Java