Experiment No.11

**Name:** Vaishnavi Santosh NaLawade.

**Roll No:**s538

**Title:** Implement and apply Strategy Design pattern for simple Shopping Cart where three payment strategies are used such as Credit Card, PayPal, BitCoin. Create the interface for strategy pattern and give concrete implementation for payment.

**Program code:**

# PaymentStrategy.java

package com.journaldev.design.strategy; public interface PaymentStrategy {

public void pay(int amount);

}

# CreditCardStrategy.java

package com.journaldev.design.strategy;

public class CreditCardStrategy implements PaymentStrategy { private String name;

private String cardNumber; private String cvv;

private String dateOfExpiry;

public CreditCardStrategy(String nm, String ccNum, String cvv, String expiryDate){ this.name=nm;

this.cardNumber=ccNum; this.cvv=cvv; this.dateOfExpiry=expiryDate;

}

@Override

public void pay(int amount) {

System.out.println(amount +" paid with credit/debit card");

}

}

# PaypalStrategy.java

package com.journaldev.design.strategy;

public class PaypalStrategy implements PaymentStrategy { private String emailId;

private String password;

public PaypalStrategy(String email, String pwd){ this.emailId=email;

this.password=pwd;

}

@Override

public void pay(int amount) {

System.out.println(amount + " paid using Paypal.");

}

}

# Item.java

package com.journaldev.design.strategy; public class Item {

private String upcCode; private int price;

public Item(String upc, int cost){ this.upcCode=upc; this.price=cost;

}

public String getUpcCode() { return upcCode;

}

public int getPrice() {

return price;

}

}

# ShoppingCart.java

package com.journaldev.design.strategy; import java.text.DecimalFormat;

import java.util.ArrayList; import java.util.List;

public class ShoppingCart {

//List of items List<Item> items;

public ShoppingCart(){

this.items=new ArrayList<Item>();

}

public void addItem(Item item){

this.items.add(item);

}

public void removeItem(Item item){ this.items.remove(item);

}

public int calculateTotal(){ int sum = 0;

for(Item item : items){

sum += item.getPrice();

}

return sum;

}

public void pay(PaymentStrategy paymentMethod){ int amount = calculateTotal(); paymentMethod.pay(amount);

}

}

# ShoppingCartTest.java

package com.journaldev.design.strategy; public class ShoppingCartTest {

public static void main(String[] args) { ShoppingCart cart = new ShoppingCart();

Item item1 = new Item("1234",10); Item item2 = new Item("5678",40);

cart.addItem(item1); cart.addItem(item2);

//pay by paypal

cart.pay(new PaypalStrategy("[myemail@example.com"](mailto:myemail@example.com), "mypwd"));

"12/15"));

}

//pay by credit card

cart.pay(new CreditCardStrategy("Pankaj Kumar", "1234567890123456", "786",

}

# Output:

50 paid using Paypal.

50 paid with credit/debit ca