```java
import java.util.ArrayList;

import java.util.List;


// Item class

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;

    }

}


// PaymentStrategy interface

public interface PaymentStrategy {

    void pay(int amount);

}


// PaypalStrategy class

public class PaypalStrategy implements PaymentStrategy {

    private String emailId;

    private String password;
```

```java
    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.");

    }

}


// CreditCardStrategy class
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.");

    }

}
```

```java
// BitcoinStrategy class
public class BitcoinStrategy implements PaymentStrategy {

    private String bitcoinAddress;


    public BitcoinStrategy(String bitcoinAddress) {

        this.bitcoinAddress = bitcoinAddress;

    }


    @Override

    public void pay(int amount) {

        System.out.println(amount + " paid using Bitcoin to address: " + bitcoinAddress);

    }

}


// ShoppingCart class
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);

    }
```

```java
    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 class

public class ShoppingCartTest {

    public static void main(String[] args) {

        ShoppingCart cart = new ShoppingCart();


        Item item1 = new Item("1234", 60);

        Item item2 = new Item("5678", 40);


        cart.addItem(item1);

        cart.addItem(item2);


        // Pay by PayPal

        cart.pay(new PaypalStrategy("myemail@example.com", "pwd"));


        // Pay by Credit Card

        cart.pay(new CreditCardStrategy("Aparna Bindage", "1234567890123456", "786", "12/15"));
```

```
        // Pay by Bitcoin
        cart.pay(new BitcoinStrategy("1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa"));
    }
}
```

Output:

100 paid using PayPal.

100 paid with credit/debit card.

100 paid using Bitcoin to address: 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa