

Java Lab 7

Anush Verma
2348504

```
import java.util.*;
class Customer{
    static int customerCount=0;
    int customerID;
    String customerName;
    String customerAddress;
    int customerPhone;
    HashSet<String> productAssociated;
    Customer(String name, String add,int pno){
        this.customerID=customerCount+1;
        this.customerName=name;
        this.customerAddress=add;
        this.customerPhone=pno;
        customerCount+=1;
        productAssociated=new HashSet<String>();
        System.out.println("Customer added");
    }
    int getCID(){
        return customerID;
    }
}
class Product{
    static int productCount=0;
    int productID;
    String productName;
    int price;
    int qty;
    Product(String name,int price,int qty){
        this.productID=productCount+1;
        this.productName=name;
        this.price=price;
        this.qty=qty;
        productCount+=1;
        System.out.println("Product Created, ID is:"+this.productID);
    }
    int getPID(){
        return productID;
    }
}
class Order{
    static int orderCount=0;
    int orderID;
```

```

    int cusID;
    HashMap<Integer,Integer> orderDetails;
    int totalAmount;
    String orderStatus;
    Order(int cid,int[] pid,int[] qty,int amt,String str){
        this.totalAmount=amt;
        this.cusID=cid;
        this.orderID=orderCount+1;
        this.orderDetails=new HashMap<Integer,Integer>();
        for(int i=0;i<pid.length;i++){
            this.orderDetails.put(pid[i],qty[i]);
        }
        this.orderStatus=str;
        orderCount+=1;
        System.out.println("Order ID="+orderID);
    }
}

class productCompare implements Comparator<Product>{
    public int compare(Product p1,Product p2){
        return p1.getPID()-p2.getPID();
    }
}

class customerCompare implements Comparator<Customer>{
    public int compare(Customer c1,Customer c2){
        return c1.getCID()-c2.getCID();
    }
}

class AmazonEcoSystem{
    TreeSet<Customer> customerList;
    TreeSet<Product> productList;
    ArrayList<Order> orderList;
    AmazonEcoSystem(){
        this.customerList=new TreeSet<Customer>(new customerCompare());
        this.productList=new TreeSet<Product>(new productCompare());
        this.orderList=new ArrayList<Order>();
    }
    void addProduct(Product p){
        productList.add(p);
    }
    int addCustomer(Customer c){
        customerList.add(c);
        return c.customerID;
    }
    void addOrder(Order a){
        orderList.add(a);
    }
    void updateProduct(int pid){
        Scanner sc=new Scanner(System.in);

```

```

int flag=0;
for(Product p:productList){
    if(p.getPID()==pid){
        System.out.println("Product ID:"+p.productID);
        System.out.println("Product Name:"+p.productName);
        System.out.println("Product Price:"+p.price);
        System.out.println("Product Qty:"+p.qty);
        System.out.println("_____");
        System.out.println("Enter new price:");
        p.price=sc.nextInt();
        System.out.println("Enter new Quantity:");
        p.qty=sc.nextInt();
        flag=1;
    }
}
if(flag==0){
    System.out.println("Product not found");
}
}

void updateCustomer(int cid){
    Scanner sc=new Scanner(System.in);
    for(Customer c:customerList){
        if(c.customerID==cid){
            System.out.println("Name:"+c.customerName);
            System.out.println("Address:"+c.customerAddress);
            System.out.println("Phone No.:"+c.customerPhone);
            System.out.println("_____");
            System.out.println("Enter new address");
            c.customerAddress=sc.nextLine();
            System.out.println("Details Updated");
        }
    }
}

void viewInventory(){
    if(productList.isEmpty()){
        System.out.println("No Products Added yet");
    }
    else{
        for(Product p:productList){
            System.out.println("Product ID:"+p.productID);
            System.out.println("Product Name:"+p.productName);
            System.out.println("Product Price:"+p.price);
            System.out.println("Product Qty:"+p.qty);
            System.out.println("_____");
        }
    }
}

void getProducts(){

```

```

        System.out.println("\nChoose Product");
        System.out.println("ID    Product    Price");
        if(productList.isEmpty()){
            System.out.println("No Products are available");
        }
        else{
            for(Product p:productList){
                if(p.qty>0){
                    System.out.println(p.productID+"    "+p.productName+"    "+
p.price);
                }
            }
        }
    }
    int getAmount(int pid,int qty){
        for(Product p:productList){
            if(p.productID==pid && p.qty>=qty){
                p.qty-=qty;
                return p.price*qty;
            }
        }
        return 0;
    }
    String getPName(int id){
        for(Product p:productList){
            if(p.productID==id){
                return p.productName;
            }
        }
        return null;
    }
    void addPname(int id,String name){
        for(Customer c:customerList){
            if(c.customerID==id){
                c.productAssociated.add(name);
            }
        }
    }
    void placeOrder(int id){
        int flag=0;
        int amount=0;
        int condition=0;
        int[] pid=new int[10];
        int[] qty=new int[10];
        Order newOrder;
        Scanner sc=new Scanner(System.in);
        int i=0;
        do

```

```

{
    System.out.println("Enter Product ID:");
    pid[i]=sc.nextInt();
    String name=getPName(pid[i]);
    addPname(id,name);
    System.out.println("Enter Qty:");
    qty[i]=sc.nextInt();
    int temp=getAmount(pid[i],qty[i]);
    if(temp==0){
        condition=1;
    }
    else{
        amount+=temp;
    }
    System.out.println("Do you want add another product to your order:
1.Yes 2.No");
    flag=sc.nextInt();
    i++;
}while(flag!=2);
if(condition==1){
    System.out.println("One or more items out of stock, Order
cancelled");
    newOrder=new Order(id,pid,qty,amount,"Cancelled");
}
else{
    newOrder=new Order(id,pid,qty,amount,"Completed");
}
addOrder(newOrder);
}
void getOrders(int id){
    for(Order o:orderList){
        if(o.cusID==id){
            System.out.println("Order ID:"+o.orderID);
            System.out.println("Amount:"+o.totalAmount);
            System.out.println("Status:"+o.orderStatus);
            System.out.println("_____");
        }
    }
}
void previousPurchases(int cid){
    System.out.println("Your Previous Purchases were: ");
    for(Customer c:customerList){
        if(c.customerID==cid){
            Iterator<String> itr=c.productAssociated.iterator();
            while(itr.hasNext()){
                System.out.print(itr.next()+" ");
            }
        }
    }
}

```



```

        break;
    case 4: aes.viewOrders();
        break;
    default: System.out.println("Invalid choice");
        break;
    }
    System.out.println("Do you want to continue as
Admin: 1.Yes 2.No");
    b=sc.nextInt();
}while(b!=2);
break;
case 2: int cid;
    System.out.println("Are you:\n1.Existing
Customer\n2.New Registration");
    int option=sc.nextInt();
    if(option==2){
        System.out.println("Enter name:");
        String name=sc.nextLine();
        sc.nextLine();
        System.out.println("Enter Address:");
        String add=sc.nextLine();
        System.out.println("Enter phone number:");
        int pno=sc.nextInt();
        Customer cus=new Customer(name,add,pno);
        cid=aes.addCustomer(cus);
    }
    else{
        System.out.println("Enter Customer ID");
        cid=sc.nextInt();
    }
    int c=0;
    do
    {
        System.out.println("Choose an operation:\n1.Place
Order\n2.Edit Address Details\n3.View All orders");
        int choice1=sc.nextInt();
        switch(choice1)
        {
            case 1: aes.previousPurchases(cid);
                aes.getProducts();
                aes.placeOrder(cid);
                break;
            case 2: aes.updateCustomer(cid);
                break;
            case 3: aes.getOrders(cid);
                break;
            default: System.out.println("Invalid Choice");
                break;
        }
    }

```

```
        }
        System.out.println("Do you want to continue as
customer: 1.Yes 2.No");
        c=sc.nextInt();
        }while(c!=2);
        break;
        default: System.out.println("Invalid Choice");
        break;
    }
    System.out.println("Do you want to: 1.Continue 2.Exit");
    a=sc.nextInt();
    }while(a!=2);
    System.out.println("Thank you");
}
}
```


Output

PS C:\Users\anush\Desktop\Christ\Java\Labs\Lab 7> java OrderManagementSystem

Choose user type:

1.Admin

2.Customer

1

Choose an operation:

1.Add product

2.Update product

3.View Inventory

4.View Orders

1

Enter Name of the product:

sadfsd

Enter Price of the Product:

46

Enter Quantity:

5

Product Created, ID is:1

Do you want to continue as Admin: 1.Yes 2.No

2

Do you want to: 1.Continue 2.Exit

1

Choose user type:

1.Admin

2.Customer

2

Are you:

1.Existing Customer

2.New Registration

2

Enter name:

sdf

Enter Address:

dfdg

Enter phone number:

15458687

Customer added

Choose an operation:

1.Place Order

2.Edit Address Details

3.View All orders

1

Your Previous Purchases were:

Choose Product

ID	Product	Price
----	---------	-------

1	sadfsd	46
---	--------	----

Enter Product ID:

1

Enter Qty:

2

Do you want add another product to your order: 1.Yes 2.No

2

Order ID=1

Do you want to continue as customer: 1.Yes 2.No

1

Choose an operation:

1.Place Order

2.Edit Address Details

3.View All orders

3

Order ID:1

Amount:92

Status:Completed

Do you want to continue as customer: 1.Yes 2.No

2

Do you want to: 1.Continue 2.Exit

2

Thank you