

welcome to E\_Commerce System!

please enter you ID:

23012132

please enter your name:

Maryam Ibrahim Mahmoud

please enter your address:

Smouha

How many products do you want to order?

4

which product would you like to add?

1-Smartphone, 2-T-Shirt, 3-OOP

1

which product would you like to add?

1-Smartphone, 2-T-Shirt, 3-OOP

1

which product would you like to add?

1-Smartphone, 2-T-Shirt, 3-OOP

2

which product would you like to add?

1-Smartphone, 2-T-Shirt, 3-OOP

3

you're total is1259.78\$

would you like to place an order? 1-yes, 2-no, any other number to end shopping

2

Do you want to delete an item in your cart?

1-yes, 2- no

1

please enter the product you want to delete:

1-Smartphone, 2-T-Shirt, 3-OOP

1

Do you want to add another product?

1-yes, 2-no

1

which product would you like to add?

1-Smartphone, 2-T-Shirt, 3-OOP

3

you're total is699.87\$

would you like to place an order? 1-yes, 2-no, any other number to end shopping

1

here's your order summary

Order Id: 1

Customer Id: 23012132

products details:

Product name=Smartphone, price=599.9\$

Product name=T-shirt, price=19.99\$

Product name=oop, price=39.99\$

Product name=oop, price=39.99\$

Total order price: 699.87\$

delivery info:

Name:Maryam Ibrahim Mahmoud

Address:Smouha

```
class Product {  
    private int productId;  
    private float price;  
    private String name;  
  
    public int getProductId() {  
        return productId;  
    }  
  
    public void setProductId(int productId) {  
        this.productId = Math.abs(productId);  
    }  
  
    public float getPrice() {  
        return price;  
    }  
  
    public void setPrice(float price) {  
        this.price = Math.abs(price);  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public Product() {  
    }  
  
    public Product (int productId, float price, String name) {  
        this.productId = Math.abs(productId);  
        this.price = Math.abs(price);  
        this.name = name;  
    }  
  
    @Override  
    public String toString() {  
        return " Product name=" + name + ", price=" + price + "$";  
    }  
}
```



```
class ElectronicProduct extends Product{
    private String brand;
    private int warrantyPeriod;

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public int getWarrantyPeriod() {
        return warrantyPeriod;
    }

    public void setWarrantyPeriod(int warrantyPeriod) {
        this.warrantyPeriod = Math.abs(warrantyPeriod);
    }

    public ElectronicProduct(String brand, int warrantyPeriod) {
        this.brand = brand;
        this.warrantyPeriod = warrantyPeriod;
    }

    public ElectronicProduct(String brand, int warrantyPeriod, int productId, float price, String name) {
        super(productId, price, name);
        this.brand = brand;
        this.warrantyPeriod = Math.abs(productId);
    }
}
```

```
class ClothingProduct extends Product{
    private String size;
    private String fabric;

    public String getSize() {
        return size;
    }

    public void setSize(String size) {
        this.size = size;
    }

    public String getFabric() {
        return fabric;
    }

    public void setFabric(String fabric) {
        this.fabric = fabric;
    }

    public ClothingProduct(String size, String fabric) {
        this.size = size;
        this.fabric = fabric;
    }

    public ClothingProduct(String size, String fabric, int productId, float price, String name) {
        super(productId, price, name);
        this.size = size;
        this.fabric = fabric;
    }
}
```

```
class BookProduct extends Product{
    private String author;
    private String publisher;

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getPublisher() {
        return publisher;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }

    public BookProduct(String author, String publisher) {
        this.author = author;
        this.publisher = publisher;
    }

    public BookProduct(String author, String publisher, int productId, float price, String name) {
        super(productId, price, name);
        this.author = author;
        this.publisher = publisher;
    }
}
```



```
class customerId{  
    private int customerId;  
  
    public int getCustomerId() {  
        return customerId;  
    }  
  
    public void setCustomerId(int customerId) {  
        this.customerId = Math.abs(customerId);  
    }  
  
    public customerId() {  
    }  
  
    public customerId(int customerId) {  
        this.customerId = Math.abs(customerId);  
    }  
  
}
```

```
class customer extends customerId{
    private String name;
    private String address;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public customer() {
    }

    public customer(String name, String address, int customerId) {
        super(customerId);
        this.name = name;
        this.address = address;
    }
}
```

```
class cart extends customerId{
    private int nProducts;
    private Product [] products;
    private int counter;

    public cart() {
        counter=0;
    }

    public cart( int customerId, int nProducts) {
        super(customerId);
        this.nProducts=Math.abs(nProducts);
        this.products=new Product[this.nProducts];
        counter=0;
    }

    public int getnProducts() {
        return nProducts;
    }

    public void setnProducts(int nProducts) {
        this.nProducts = Math.abs(nProducts);
    }

    public void setProducts(Product[] products) {
        this.products = products;
    }

    public Product[] getProducts() {
        return products;
    }

    public void addProduct(Product product){
        if(counter<nProducts){
            products[counter]= product;
            counter ++;
        }else
            System.out.println("cart is full");
    }
}
```



```

public Product[] addProduct2(Product[] products, Product product) {
    Product [] products2=new Product[products.length+1];
    int i;
    for(i=0;i<products.length;i++){
        products2[i]=products[i];
    }
    products2[i]=product;
    setProducts(products2);
    setnProducts(products2.length);
    return getProducts();
}

public Product[] removeProduct(Product[] products, Product product) {//////////
    if (products == null || products.length == 0) {
        return products;
    }else{
        Product [] products2=new Product[products.length-1];
        int j=0;
        boolean found=false;
        for(int i=0;i<products.length;i++){
            if(!found&&products[i]==product){
                found=true;
            }else{
                products2[j]=products[i];
                j++;
            }
        }
        setProducts(products2);
        setnProducts(products2.length);
        return products2;
    }
}

```

```
public boolean inTheOrder(Product product){
    for(Product i:products){
        if(i!=null && i.getName().equals(product.getName()) && i.getProductid()==product.getProductid())
            return true;
    }
    return false;
}

public float calculateprice(){
    float sum =0;
    for(Product i:products){
        if(i!=null)
            sum+= i.getPrice();
    }
    return sum;
}

public Order placeOrder(int oId , int customerId){
    Order order=new Order(oId,products,customerId);
    return order;
}
```



```
class Order extends cart{
    private int orderId;
    private float totalPrice;

    public Order() {
    }

    public Order(int orderId, Product[]products, int customerId) {
        super.setCustomerId(customerId);
        this.orderId = orderId;
        super.setProducts(products);
    }

    public int getOrderId() {
        return orderId;
    }

    public void setOrderId(int orderId) {
        this.orderId = orderId;
    }

    public float getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(cart cart) {
        this.totalPrice = cart.calculateprice();
    }

    public void printOrderInfo(){
        System.out.println("here's your order summary");
        System.out.println("Order Id: "+ getOrderId());
        System.out.println("Customer Id: "+ getCustomerId() );
        System.out.println("products details:");
        for(Product i: getProducts()){
            System.out.println(i.toString());
        }
        System.out.println("Total order price: "+ getTotalPrice() +"$");
    }
}
```



```

public class EcommerceSystem {

    public static void main(String[] args) {
        ElectronicProduct phone =new ElectronicProduct("Samsung",1,1,599.9f,"Smartphone");
        ClothingProduct tShirt=new ClothingProduct("medium","cotton",2,19.99f,"T-shirt");
        BookProduct book=new BookProduct("O'Reilly","X Publications",3,39.99f,"oop");
        Scanner in =new Scanner(System.in);
        System.out.println("""
            welcome to E_Commerce System!
            please enter you ID:"");

        int c_Id=in.nextInt();
        in.nextLine();
        System.out.println("please enter your name:");
        String name=in.nextLine();
        System.out.println("please enter your address:");
        String address=in.nextLine();
        customer customer1=new customer(name, address,c_Id);
        System.out.println("How many products do you want to order?");
        int nProducts=in.nextInt();
        cart cart1=new cart(c_Id, nProducts);
        for(int i=0;i<cart1.getnProducts();i++){
            System.out.println("""
                which product would you like to add?
                1-Smartphone, 2-T-Shirt, 3-OOP""");
            int choice=in.nextInt();
            switch(choice){
                case 1:cart1.addProduct(phone);
                    break;
                case 2:cart1.addProduct(tShirt);
                    break;
                case 3:cart1.addProduct(book);
                    break;
                default:
                    System.out.println("invalid number choice");
                    i--;
            }
        }
    }
}

```

```

boolean flag1;
do{
    flag1=false;
    System.out.println("you're total is"+cart1.calculateprice()+"$"+
        "\n would you like to place an order? 1=yes, 2=no, any other number to end shopping");
    int decision=in.nextInt();
    switch (decision) {
        case 1:
            Order order = cart1.placeOrder(1, c_Id);
            order.setTotalPrice(cart1);
            order.printOrderInfo();
            System.out.println("delivery info:\n"+
                "Name:"+customer1.getName()+"\n"+
                "Address:"+customer1.getAddress());
            break;
        case 2:
            flag1=true;
            System.out.println("""
                Do you want to delete an item in your cart?
                1=yes, 2- no""");
            int decision2=in.nextInt();
            boolean flag2;
            do{
                flag2=false;
                if(decision2==1){
                    System.out.println("""
                        please enter the product you want to delete:
                        1-Smartphone, 2-T-Shirt, 3-OOP""");
                    int decision3=in.nextInt();/////
                    switch(decision3){
                        case 1 :
                            if(cart1.inTheOrder(phone)){
                                cart1.removeProduct(cart1.getProducts(),phone);
                                cart1.calculateprice();
                            }else{
                                System.out.println("product is not in the cart!");
                                flag2=true;
                            }
                            break;
                        case 2:
                            if(cart1.inTheOrder(tShirt)){
                                cart1.removeProduct(cart1.getProducts(),tShirt);
                                cart1.calculateprice();
                            }else{
                                System.out.println("product is not in the cart!");

```



```
        case 2:
            if(cart1.inTheOrder(tShirt)){
                cart1.removeProduct(cart1.getProducts(),tShirt);
                cart1.calculateprice();
            }else{
                System.out.println("product is not in the cart!");
                flag2=true;
            }
            break;
        case 3 :
            if(cart1.inTheOrder(book)){
                cart1.removeProduct(cart1.getProducts(),book);
                cart1.calculateprice();
            }else{
                System.out.println("product is not in the cart!");
                flag2=true;
            }
            break;
        default: System.out.println("invalid number choice");
                flag2=true;
    }
}
```



```

    }while(flag2);
    System.out.println("""
        Do you want to add another product?
        1-yes, 2-no""");
    int decision4=in.nextInt();
    if(decision4==1){
        boolean flag3;
        do{
            flag3=false;
            System.out.println("""
                which product would you like to add?
                1-Smartphone, 2-T-Shirt, 3-OOP""");
            int choice=in.nextInt();
            switch(choice){
                case 1: cart1.addProduct2(cart1.getProducts(),phone);
                        cart1.calculateprice();
                        break;
                case 2 : cart1.addProduct2(cart1.getProducts(),tShirt);
                        cart1.calculateprice();
                        break;
                case 3 : cart1.addProduct2(cart1.getProducts(),book);
                        cart1.calculateprice();
                        break;
                default : System.out.println("invalid number choice, please enter a proper number.");
                        flag3=true;
            }
        }while(flag3);
    }
    break;
    default:
        System.out.println("we'll be waiting for your return!");
        break;
}
}while(flag1);
}
}

```

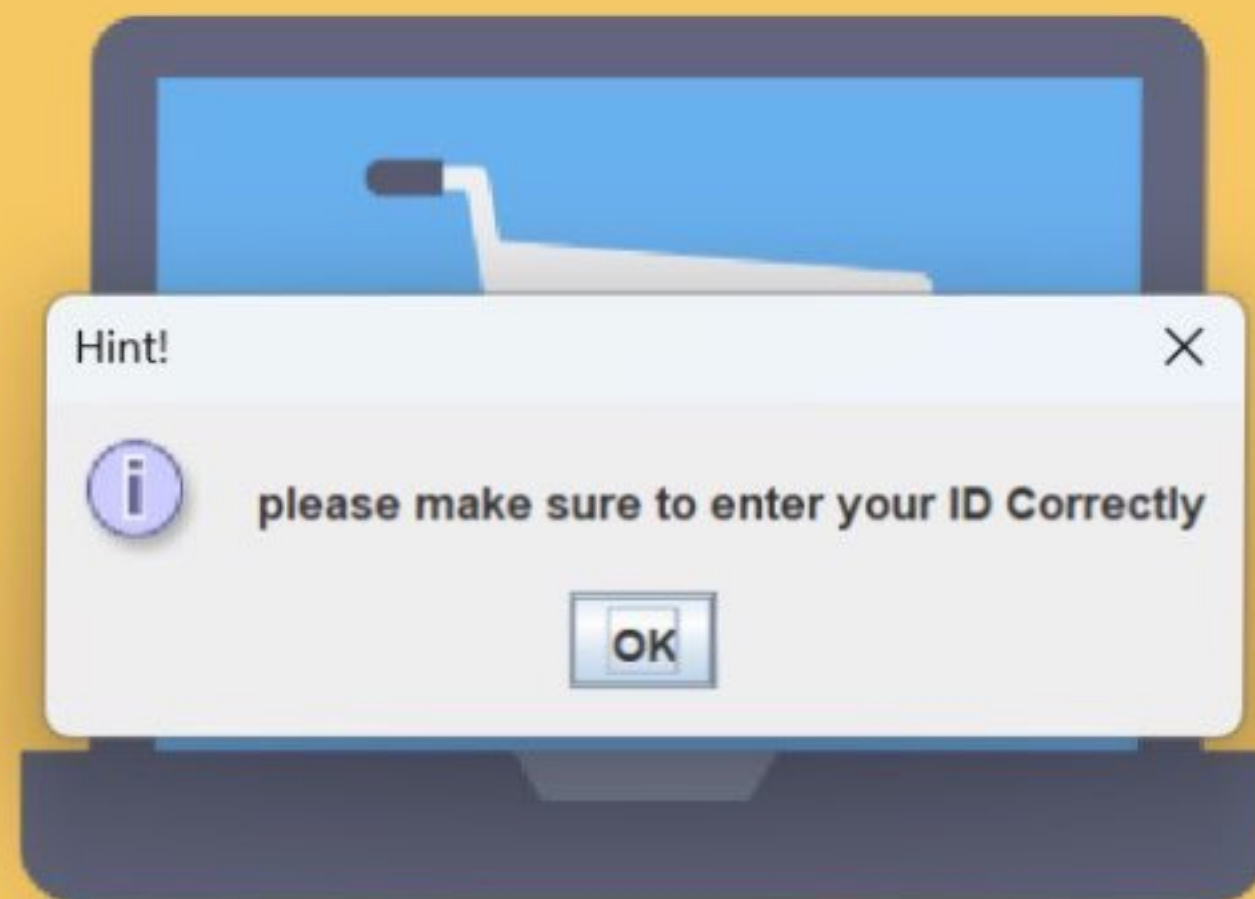


**Welcome to E-Commerce System! please enter your :**

ID

Name

Address



**Welcome to E-Commerce System! please enter your :**

ID

Maryam Ebrahim Mahmoud

Smouha

[Start Shopping ->](#)





**Welcome to E-Commerce System! please enter your :**



**please enter the number of products you're willing to add to your cart:**

[Start adding to cart->](#)



**select products to add to your cart:**

Product name=Smartphone, price=599.9\$ ▼

Product name=Smartphone, price=599.9\$ ▼

Product name=Smartphone, price=599.9\$ ▼

Product name=Smartphone, price=599.9\$ ▼

Place an order

Product name=Smartphone, price=599.9\$

Product name=T-shirt, price=19.99\$

Product name=Teen book, price=20.00\$

40

```
customerName= new JTextField();
```

41

```
customerID= new JTextField();
```

42

```
cAddress= new JTextField();
```





**Thanks for shopping at E-Commerce System!**

**Order Summary :**

**Product name=Smartphone, price=599.9\$ total price = 679.87\$**

**Product name=T-shirt, price=19.99\$**

**Customer name: Maryam Ibrahim Mahmoud**

**Product name=T-shirt, price=19.99\$**

**Customer ID: 23012132**

**Product name=oop book, price=39.99\$**

**Address: Smouha**



```
1 package e_commerce;
2 import javax.swing.*.*;
3 import java.awt.*.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import javax.swing.ImageIcon;
7 import javax.swing.border.Border;
8
9
10 public class StartUpPage extends JFrame implements ActionListener{
11     JButton start;
12     JLabel intro;
13     ImageIcon logo;
14     JTextField customerID, customerName, cAddress;
15     String name,address;
16     int ID;
17
18
19     StartUpPage(){
20
21
22         logo =new ImageIcon("e-c.png");
23         Border logoBorder =BorderFactory.createLineBorder(Color.ORANGE,10);
24
25         intro=new JLabel();
26         intro.setText("Welcome to E-Commerce System! "+"\\n"
27             +" please enter your :");
28         intro.setIcon(logo);
29         intro.setHorizontalTextPosition(JLabel.CENTER);
30         intro.setVerticalTextPosition(JLabel.BOTTOM);
31         intro.setIconTextGap(-100);
32         intro.setFont(new Font("Bernard MT Condensed", Font.BOLD, 30));
33         intro.setBorder(logoBorder);
34         intro.setHorizontalAlignment(JLabel.CENTER);
35         intro.setVerticalAlignment(JLabel.CENTER);
36         intro.setBounds(100, 100, 150, 150);
37
38
39         customerName= new JTextField();
40         customerID= new JTextField();
41         cAddress=new JTextField();
42
43         customerName.setText("Name");
44         cAddress.setText("Address");
45         customerID.setText("ID");
46
47         customerName.setPreferredSize(new Dimension(250,40));
48         customerID.setPreferredSize(new Dimension(250,40));
49         cAddress.setPreferredSize(new Dimension(250,40));
50
```



```
51
52     start= new JButton("Start Shopping ->");
53     start.addActionListener(this);
54
55     JPanel textBoxPanel = new JPanel();
56     textBoxPanel.add(customerID);
57     textBoxPanel.add(customerName);
58     textBoxPanel.add(cAddress);
59
60     JPanel buttonPanel = new JPanel();
61     buttonPanel.add(textBoxPanel);
62     buttonPanel.add(start);
63
64     this.setTitle("E-Commerce System");
65     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
66     this.setIconImage(logo.getImage());
67     this.getContentPane().setBackground(Color.WHITE);
68     this.setResizable(false);
69     this.setLayout(new BorderLayout());
70     this.add(intro, BorderLayout.CENTER);
71     this.add(buttonPanel, BorderLayout.SOUTH);
72     this.setSize(500, 600);
73     this.setVisible(true);
74     this.pack();
75
76
77
78 }
79 public void actionPerformed(ActionEvent e){
80
81     if(e.getSource()==start){
82
83
84         if(customerID.getText().equals("ID")) {
85             JOptionPane.showMessageDialog(this, "please make sure to enter your ID Correctly",
86                 "Hint!", JOptionPane.INFORMATION_MESSAGE);
87         }if(customerName.getText().equals("Hint!")) {
88             JOptionPane.showMessageDialog(this, "please make sure to enter your name",
89                 "error", JOptionPane.INFORMATION_MESSAGE);
90         }if(cAddress.getText().equals("Address")) {
91             JOptionPane.showMessageDialog(this, "please make sure to enter your Address",
92                 "Hint!", JOptionPane.INFORMATION_MESSAGE);
93         }else {
94             name=customerName.getText();
95             address=cAddress.getText();
96             ID=Math.abs(Integer.parseInt(customerID.getText()));
97             new Items(this.address,this.name,this.ID);
98             this.dispose();
99         }
100
101
102 }
```



```
1 package e_commerce;
2 import javax.swing.*;
3
4
5
6
7
8
9 public class Items extends JFrame implements ActionListener {
10     public int nProducts;
11     JTextField nProductsIN;
12     JButton next;
13     ImageIcon logo;
14     JLabel page2;
15     String name,address;
16     int ID;
17
18     int getNProducts() {
19         return nProducts;
20     }
21
22     Items(String address,String name, int ID){
23
24         this.address=address;
25         this.name=name;
26         this.ID=ID;
27
28         logo =new ImageIcon("e-c.png");
29         Border logoBorder =BorderFactory.createLineBorder(Color.ORANGE,10);
30
31         page2= new JLabel();
32         page2.setText("please enter the number of products you're willing to add to your cart:");
33         page2.setIcon(logo);
34         page2.setHorizontalTextPosition(JLabel.CENTER);
35         page2.setVerticalTextPosition(JLabel.BOTTOM);
36         page2.setIconTextGap(-100);
37         page2.setFont(new Font("Bernard MT Condensed", Font.BOLD, 25));
38         page2.setBorder(logoBorder);
39         page2.setHorizontalAlignment(JLabel.CENTER);
40         page2.setVerticalAlignment(JLabel.CENTER);
41         page2.setBounds(100, 100, 150, 150);
42
43
44         nProductsIN = new JTextField();
45         nProductsIN.setPreferredSize(new Dimension(250,40));
46
47
48         next = new JButton("Start adding to cart->");
49         next.addActionListener(this);
50
51         JPanel bottomPanel = new JPanel();
52         bottomPanel.add(nProductsIN);
53         bottomPanel.add(next);
```



```

54
55     this.setTitle("E-Commerce System");
56     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
57     this.setIconImage(logo.getImage());
58     this.getContentPane().setBackground(Color.WHITE);
59     this.setResizable(false);
60     this.setLayout(new BorderLayout());
61     this.setVisible(true);
62     this.add(page2, BorderLayout.CENTER);
63     this.add(bottomPanel, BorderLayout.SOUTH);
64     this.pack();
65
66 }
67 public void actionPerformed(ActionEvent e){
68     if(e.getSource()==next){
69         nProducts=Math.abs(Integer.parseInt(nProductsIN.getText()));
70         new Window2(this.getNProducts(),this.address,this.name,this.ID);
71
72         this.dispose();
73     }
74 }
75 }
76 }
77 }
78

```



```
1 package e_commerce;
2 import javax.swing.*;
3
4
5
6
7
8
9 public class Window2 extends JFrame implements ActionListener {
10     JComboBox[] products;
11     int nProducts;
12     JButton next;
13     ImageIcon logo;
14     JLabel page2;
15     JPanel panel, panel2;
16     Product[] productsArr;
17     Product[] selectedProducts;
18     String name, address;
19     int ID;
20
21     public Window2(int nProducts, String address, String name, int ID) {
22         this.nProducts = nProducts;
23         this.address = address;
24         this.name = name;
25         this.ID = ID;
26
27         logo = new ImageIcon("e-c.png");
28         Border logoBorder = BorderFactory.createLineBorder(Color.ORANGE, 10);
29
30         page2 = new JLabel();
31         page2.setText("select products to add to your cart:");
32         page2.setIcon(logo);
33         page2.setHorizontalTextPosition(JLabel.CENTER);
34         page2.setVerticalTextPosition(JLabel.BOTTOM);
35         page2.setIconTextGap(-100);
36         page2.setFont(new Font("Bernard MT Condensed", Font.BOLD, 25));
37         page2.setBorder(logoBorder);
38         page2.setHorizontalAlignment(JLabel.CENTER);
39         page2.setVerticalAlignment(JLabel.CENTER);
40         page2.setBounds(100, 100, 150, 150);
41
42         next = new JButton("Place an order");
43         next.addActionListener(this);
44
45         ElectronicProduct phone = new ElectronicProduct("Samsung", 1, 1, 599.9f, "Smartphone");
46         ClothingProduct tShirt = new ClothingProduct("medium", "cotton", 2, 19.99f, "T-shirt");
47         BookProduct book = new BookProduct("O'Reilly", "X Publications", 3, 39.99f, "oop book");
48         productsArr = new Product[3];
49         productsArr[0] = phone;
50         productsArr[1] = tShirt;
51         productsArr[2] = book;
52
```



```

products = new JComboBox [nProducts];
selectedProducts = new Product[nProducts];
panel = new JPanel();
for (int i = 0; i < nProducts; i++) {
    products[i] = new JComboBox<>(productsArr);
    products[i].addActionListener(this);
    panel.add(products[i]);
}

panel2 = new JPanel();
panel2.add(panel);
panel2.add(next);

this.setTitle("E-Commerce System");
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setIconImage(logo.getImage());
this.getContentPane().setBackground(Color.WHITE);
this.setResizable(false);
this.setLayout(new BorderLayout());
this.add(page2, BorderLayout.CENTER);
this.add(panel2, BorderLayout.SOUTH);
this.pack();
this.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    for (int i = 0; i < nProducts; i++) {
        if (e.getSource() == products[i]) {
            selectedProducts[i] = (Product) products[i].getSelectedItem();
        }
    }

    if (e.getSource() == next) {
        this.dispose();
        new Order(this.selectedProducts, this.address, this.name, this.ID);
    }
}
}
}

```



```
1 package e_commerce;
2 import javax.swing.*;
3
4
5
6
7
8
9 public class Order extends JFrame {
10     Object[] selectedProducts;
11     String name,address;
12     int ID;
13
14     Order(Product [] selectedProducts,String address,String name, int ID){
15         this.address=address;
16         this.name=name;
17         this.ID=ID;
18
19         ImageIcon logo = new ImageIcon("e-c.png");
20         Border logoBorder = BorderFactory.createLineBorder(Color.ORANGE, 10);
21         JLabel page4= new JLabel();
22         page4 = new JLabel();
23         page4.setText("Thanks for shopping at E-Commerce System! ");
24         page4.setIcon(logo);
25         page4.setHorizontalTextPosition(JLabel.CENTER);
26         page4.setVerticalTextPosition(JLabel.BOTTOM);
27         page4.setIconTextGap(-100);
28         page4.setFont(new Font("Bernard MT Condensed", Font.BOLD, 25));
29         page4.setBorder(logoBorder);
30         page4.setHorizontalAlignment(JLabel.CENTER);
31         page4.setVerticalAlignment(JLabel.CENTER);
32         page4.setBounds(100, 100, 150, 150);
33
34
35         JLabel title =new JLabel("Order Summary :");
36         title.setFont(new Font("Bernard MT Condensed", Font.BOLD, 25));
37
38         JPanel panel = new JPanel();
39         panel.setLayout(new GridLayout(selectedProducts.length, 1));
40         float totalPrice=0f;
41         for (Product product : selectedProducts) {
42             JLabel productLabel = new JLabel(product.toString());
43             productLabel.setHorizontalAlignment(SwingConstants.LEFT);
44             productLabel.setFont(new Font("Bernard MT Condensed", Font.LAYOUT_LEFT_TO_RIGHT, 20));
45             totalPrice+=product.getPrice();
46             panel.add(productLabel);
47         }
48         JTextArea orderDetails= new JTextArea("total price = "+totalPrice+"$ \n"+
49             "Customer name: "+this.name+"\n"+
50             "Customer ID: "+this.ID+"\n"+
51             "Address: "+this.address);
52         orderDetails.setFont(new Font("Bernard MT Condensed", Font.LAYOUT_LEFT_TO_RIGHT, 20));
53         orderDetails.setBackground(null);
54     }
```



```
JPanel panel2 = new JPanel();
    panel2.add(title);
    panel2.add(panel);
    panel2.add(orderDetails);

this.setTitle("E-Commerce System");
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setIconImage(logo.getImage());
this.getContentPane().setBackground(Color.WHITE);
this.setResizable(false);
this.setLayout(new BorderLayout());
this.add(panel2, BorderLayout.SOUTH);
this.setVisible(true);
this.add(page4, BorderLayout.CENTER);
this.pack();
```

```
}
```



```
1 package e_commerce;
2
3 class Product{
4     private int productId;
5     private float price;
6     private String name;
7
8     public int getProductId() {
9         return productId;
10    }
11
12    public void setProductId(int productId) {
13        this.productId = Math.abs(productId);
14    }
15
16    public float getPrice() {
17        return price;
18    }
19
20    public void setPrice(float price) {
21        this.price = Math.abs(price);
22    }
23
24    public String getName() {
25        return name;
26    }
27
28    public void setName(String name) {
29        this.name = name;
30    }
31
32    public Product() {
33    }
34
35    public Product (int productId, float price, String name) {
36        this.productId = Math.abs(productId);
37        this.price = Math.abs(price);
38        this.name = name;
39    }
40
41    @Override
42    public String toString() {
43        return " Product name=" + name + ", price=" + price + "$";
44    }
45
46
47 }
48
```

```
49
50 class ElectronicProduct extends Product{
51     private String brand;
52     private int warrantyPeriod;
53
54     public String getBrand() {
55         return brand;
56     }
57
58     public void setBrand(String brand) {
59         this.brand = brand;
60     }
61
62     public int getWarrantyPeriod() {
63         return warrantyPeriod;
64     }
65
66     public void setWarrantyPeriod(int warrantyPeriod) {
67         this.warrantyPeriod = Math.abs(warrantyPeriod);
68     }
69
70     public ElectronicProduct(String brand, int warrantyPeriod) {
71         this.brand = brand;
72         this.warrantyPeriod = warrantyPeriod;
73     }
74
75     public ElectronicProduct(String brand, int warrantyPeriod, int productId, float price, String name) {
76         super(productId, price, name);
77         this.brand = brand;
78         this.warrantyPeriod = Math.abs(productId);
79     }
80
81 }
```



```
115 class BookProduct extends Product{
116     private String author;
117     private String publisher;
118
119     public String getAuthor() {
120         return author;
121     }
122
123     public void setAuthor(String author) {
124         this.author = author;
125     }
126
127     public String getPublisher() {
128         return publisher;
129     }
130
131     public void setPublisher(String publisher) {
132         this.publisher = publisher;
133     }
134
135     public BookProduct(String author, String publisher) {
136         this.author = author;
137         this.publisher = publisher;
138     }
139
140     public BookProduct(String author, String publisher, int productId, float price, String name) {
141         super(productId, price, name);
142         this.author = author;
143         this.publisher = publisher;
144     }
145 }
146
```



```

32 class ClothingProduct extends Product{
33     private String size;
34     private String fabric;
35
36     public String getSize() {
37         return size;
38     }
39
40     public void setSize(String size) {
41         this.size = size;
42     }
43
44     public String getFabric() {
45         return fabric;
46     }
47
48     public void setFabric(String fabric) {
49         this.fabric = fabric;
50     }
51
52     public ClothingProduct(String size, String fabric) {
53         this.size = size;
54         this.fabric = fabric;
55     }
56
57     public ClothingProduct(String size, String fabric, int productId, float price, String name) {
58         super(productId, price, name);
59         this.size = size;
60         this.fabric = fabric;
61     }
62 }

```

```
1 package e_commerce;
2
3 public class E_CommerceSystem {
4
5     public static void main(String[] args) {
6
7         new StartUpPage();
8     }
9 }
10
11
12
```