

```

package ecommerceSystemm;

import java.util.Scanner;
import javax.swing.SwingUtilities;

public class EcommerceSystemm {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new GUI());
        Scanner scanner = new Scanner(System.in);

        ElectronicProduct electronicProduct = new ElectronicProduct(brand: "Samsung", warrantyPeriod: 1 , productId:1 , name: "smartphone" , (float)599.9);
        ClothingProduct clothingProduct = new ClothingProduct(size: "Medium" , fabric:"Cotton" , productId:2 , name: "T-shirt" , (float)19.99 );
        BookProduct bookProduct = new BookProduct(author:"O'Reilly" , publisher:"X Publications" , productId:3 , name: "OOP" , (float)39.99);

        // Create customer
        System.out.println(x: "Welcome to the E-Commerce System!");
        System.out.println(x: "Enter your customer ID: ");
        int customerId = scanner.nextInt();
        System.out.println(x: "Enter your name: ");
        String customerName = scanner.nextLine();
        scanner.nextLine(); // Consume extra newline character
        System.out.println(x: "Enter your address: ");
        String customerAddress = scanner.nextLine();
        Customer customer = new Customer(customerId, name: customerName, address: customerAddress);

        System.out.println(x: "How many products you want to add to your cart? ");
        int itemsnum = scanner.nextInt();
        Cart cart = new Cart(customerId , nProducts:itemsnum);

        for(int i = 0; i<itemsnum; i++){
            System.out.println(x: "Which product you would like to add? 1- Smartphone 2- T-shirt 3- OOP ");
            int choice = scanner.nextInt();
            switch(choice){
                case 1 :
                    cart.addProduct(product: electronicProduct);
                    break;
                case 2 :
                    cart.addProduct(product: clothingProduct);
                    break;
                case 3 :
                    cart.addProduct(product: bookProduct);
                    break;
            }
        }
    }
}
    
```

```
26 Cart cart = new Cart(customerId , nProducts:itemsnum);
27
28 for(int i = 0; i<itemsnum; i++){
29     System.out.println(x: "Which product you would like to add? 1- Smartphone 2- T-shirt 3- OOP ");
30     int choice = scanner.nextInt();
31     switch(choice){
32         case 1 :
33             cart.addProduct(product: electronicProduct);
34             break;
35         case 2 :
36             cart.addProduct(product: clothingProduct);
37             break;
38         case 3 :
39             cart.addProduct(product: bookProduct);
40             break;
41         default :
42             System.out.println(x: "Unavailable option");
43     }
44 }
45
46 System.out.print("The total is $" + cart.calculatePrice() + " ");
47 System.out.println(x: "Would you like to place the order? 1- Yes 2- No");
48 int option = scanner.nextInt();
49 switch(option){
50     case 1 :
51         cart.placeOrder();
52         break;
53     case 2 :
54         System.out.println(x: "Thank you");
55         break;
56     default :
57         System.out.println(x: "Unavailable option");
58 }
59 }
60 }
61
```



```
1 package ecommerce.systemmm;
2
3 public class BookProduct extends Product{
4     private String author;
5     private String publisher;
6
7     BookProduct(String author, String publisher, int productId, String name, float price){
8         super(productId, name, price);
9         this.author = author;
10        this.publisher = publisher;
11    }
12
13    public String getAuthor(){
14        return author;
15    }
16
17    public String getPublisher(){
18        return publisher;
19    }
20
21    public void setAuthor(String author){
22        this.author = author;
23    }
24
25    public void setPublisher(String publisher){
26        this.publisher = publisher;
27    }
28
29 }
```




```
1 package ecommerceSystem;
2 public class Cart {
3     private int customerId;
4     private int nProducts;
5     private int element = 0;
6     private Product[] products;
7
8     Cart(int customerId, int nProducts) {
9         this.customerId = customerId;
10        this.nProducts = nProducts;
11        this.products = new Product[nProducts];
12    }
13    public int getCustomerId() {
14        return customerId;
15    }
16    public int getnProducts() {
17        return nProducts;
18    }
19
20    public Product[] getProducts() {
21        return products;
22    }
23    public int setCustomerId(int customerId) {
24        this.customerId = customerId;
25        if (customerId < 0) {
26            return Math.abs(customerId);
27        } else {
28            return customerId;
29        }
30    }
31    public int setnProducts(int nProducts) {
32        this.nProducts = nProducts;
33        if (nProducts < 0) {
34            return Math.abs(nProducts);
35        } else {
36            return nProducts;
37    }
38 }
```

```
30 this.nProducts = nProducts;
31 if(nProducts < 0){
32     return Math.abs(a: nProducts);
33 }else{
34     return nProducts;
35 }
36 }
37 public void setProducts(Product[] products){
38     this.products = products;
39 }
40 public void addProduct(Product product){
41     products[this.element] = product;
42     this.element++;
43 }
44 public void removeProduct(int index){
45     if (index >= 0 && index < nProducts){
46         for (int i = index; i < nProducts - 1; i++) {
47             products[i] = products[i + 1];
48         }
49         nProducts--;
50         products[nProducts] = null; // Set the last element to null for clarity
51     }else{
52         System.out.println(x: "Invalid product index!");
53     }
54 }
55 public float calculatePrice(){
56     float totalPrice = 0;
57     for(int i = 0; i < nProducts; i++){
58         totalPrice += products[i].getPrice();
59     }
60     return totalPrice;
61 }
62 public void placeOrder(){
63     Order order = new Order(customerId , orderId: 1 , products , (float)calculatePrice());
64     order.printOrderInfo();
65 }
66 }
```



```
package ecommercessystem;
```

```
public class ClothingProduct extends Product{
```

```
    private String size;
```

```
    private String fabric;
```

```
    ClothingProduct(String size, String fabric, int productId, String name, float price){
```

```
        super(productId, name, price);
```

```
        this.size = size;
```

```
        this.fabric = fabric;
```

```
    }
```

```
    public String getSize(){
```

```
        return size;
```

```
    }
```

```
    public String getFabric(){
```

```
        return fabric;
```

```
    }
```

```
    public void setSize(String size){
```

```
        this.size = size;
```

```
    }
```

```
    public void setFabric(String fabric){
```

```
        this.fabric = fabric;
```

```
    }
```

```
}
```



```
1 package ecommerceSystem;
2
3 public class Customer {
4     private int customerId;
5     private String name;
6     private String address;
7
8     Customer(int customerId, String name, String address){
9         this.customerId = customerId;
10        this.name = name;
11        this.address = address;
12    }
13    public int getCustomerId(){
14        return customerId;
15    }
16    public String getName(){
17        return name;
18    }
19    public String getAddress(){
20        return address;
21    }
22    public void setName(String name){
23        this.name = name;
24    }
25    public int setCustomerId(int customerId){
26        this.customerId = customerId;
27        if(customerId < 0){
28            return Math.abs(customerId);
29        }else{
30            return customerId;
31        }
32    }
33    public void setAddress(String address){
34        this.address = address;
35    }
36 }
```



```
package ecommerceSystemm;
```

```
public class ElectronicProduct extends Product {
```

```
    private String brand;
```

```
    private int warrantyPeriod;
```

```
    ElectronicProduct(String brand, int warrantyPeriod, int productId, String name, float price){
```

```
        super(productId, name, price);
```

```
        this.brand = brand;
```

```
        this.warrantyPeriod = warrantyPeriod;
```

```
    }
```

```
    public String getBrand(){
```

```
        return brand;
```

```
    }
```

```
    public int getWarrantyPeriod(){
```

```
        return warrantyPeriod;
```

```
    }
```

```
    public void setBrand(String brand){
```

```
        this.brand = brand;
```

```
    }
```

```
    public int setWarrantyPeriod(int warrantyPeriod){
```

```
        this.warrantyPeriod = warrantyPeriod;
```

```
        if(warrantyPeriod < 0){
```

```
            return Math.abs(warrantyPeriod);
```

```
        }else{
```

```
            return warrantyPeriod;
```

```
        }
```

```
    }
```

```
}
```




```

package ecommerceSystemmm;

public class Order {
    private int customerId;
    private int orderId;
    private Product[] products;
    private float totalPrice;
    Order(int customerId, int orderId, Product[] products, float totalPrice){
        this.customerId = customerId;
        this.orderId = orderId;
        this.products = products;
        this.totalPrice = totalPrice;
    }
    public int getCustomerId(){
        return customerId;
    }
    public int getOrderId(){
        return orderId;
    }
    public Product[] getProducts(){
        return products;
    }
    public double getTotalPrice(){
        return totalPrice;
    }
    public int setCustomerId(int customerId){
        this.customerId = customerId;
        if(customerId < 0){
            return Math.abs(a: customerId);
        }else{
            return customerId;
        }
    }
    public int setOrderId(int orderId){
        this.orderId = orderId;
        if(orderId < 0){
            return Math.abs(a: orderId);
        }else{

```



```
return Math.abs(a: customerId);
    }else{
return customerId;
    }
    }

    public int setOrderId(int orderId){
this.orderId = orderId;
    if(orderId <0){
return Math.abs(a: orderId);
    }else{
return orderId;
    }
    }

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

    public float setTotalPrice(float totalPrice){
this.totalPrice = totalPrice;
    if(totalPrice <0){
return Math.abs(a: totalPrice);
    }else{
return totalPrice;
    }
    }

    public void printOrderInfo(){
System.out.println(x: "Here's your order's summary:");
System.out.println("Order ID: " + orderId);
System.out.println("Customer ID: " + customerId);
System.out.println(x: "Products: " );
for(int i = 0 ; i< products.length ; i++){
System.out.println(products[i].getName() + " - $" + products[i].getPrice());
    }
System.out.println("Total price: $" + totalPrice);
    }
}
```

```
package ecommerceSystemm;
```

```
public class Product {  
    private int productId;  
    private String name;  
    private float price;  
    Product(int productId, String name, float price) {  
        this.productId = productId;  
        this.name = name;  
        this.price = price;  
    }  
    public int getProductId() {  
        return productId;  
    }  
    public String getName() {  
        return name;  
    }  
    public float getPrice() {  
        return price;  
    }  
    public int setProductId(int productId) {  
        this.productId = productId;  
        if(productId < 0) {  
            return Math.abs(productId);  
        } else {  
            return productId;  
        }  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public float setPrice(float price) {  
        this.price = price;  
        if(price < 0) {  
            return Math.abs(price);  
        } else {  
            return price;  
        }  
    }  
}
```




```

    ,
    public float getPrice() {
        return price;
    }
    public int setProductId(int productId) {
        this.productId = productId;
        if(productId < 0) {
            return Math.abs(a: productId);
        }else{
            return productId;
        }
    }
    public void setName(String name) {
        this.name = name;
    }
    public float setPrice(float price) {
        this.price = price;
        if(price < 0) {
            return Math.abs(a: price);
        }else{
            return price;
        }
    }
}
}

```

```

1 package ecommercessystem;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 public class GUI extends JFrame{
8
9     private JLabel customerIdLabel, nameLabel, addressLabel, nProductLabel, productsLabel, totalPriceLabel;
10    private JTextField customerIdField, nameField, addressField, nProductField;
11    private JComboBox<String> productsComboBox;
12    private JButton addProductButton, calculatePriceButton, printOrderButton;
13    private String[] selectedProducts = new String[4]; // Assuming a maximum of 3 products can be selected
14    private double totalPrice = 0;
15
16    public GUI() {
17        setTitle(title: "E-Commerce System");
18        setSize(width: 600, height: 400);
19        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
20        setLayout(new GridLayout(rows: 7, cols: 2));
21
22        customerIdLabel = new JLabel(text: "Enter your ID:");
23        add(comp: customerIdLabel);
24        customerIdField = new JTextField();
25        add(comp: customerIdField);
26
27        nameLabel = new JLabel(text: "Enter your Name:");
28        add(comp: nameLabel);
29        nameField = new JTextField();
30        add(comp: nameField);
31
32        addressLabel = new JLabel(text: "Enter your Address:");
33        add(comp: addressLabel);
34        addressField = new JTextField();
35        add(comp: addressField);
36
37        nProductLabel = new JLabel(text: "How many products you want to add to your cart? ");

```



```
37 nProductLabel = new JLabel(text: "How many products you want to add to your cart? ");
38 add(comp: nProductLabel);
39 nProductField = new JTextField();
40 add(comp: nProductField);
41
42 productsLabel = new JLabel(text: "Select Product:");
43 add(comp: productsLabel);
44 String[] products = {"Smartphone - $599.9", "T-shirt - $19.99", "OOP Book - $39.99"};
45 productsComboBox = new JComboBox<>(items: products);
46 add(comp: productsComboBox);
47
48 addProductButton = new JButton(text: "Add Product");
49 add(comp: addProductButton);
50
51 totalPriceLabel = new JLabel(text: "Total Price: $0.0");
52 add(comp: totalPriceLabel);
53
54 calculatePriceButton = new JButton(text: "Calculate Total Price");
55 add(comp: calculatePriceButton);
56
57 printOrderButton = new JButton(text: "Print Order Info");
58 add(comp: printOrderButton);
59
60 addProductButton.addActionListener(new ActionListener() {
61     @Override
62     public void actionPerformed(ActionEvent e) {
63         for (int i = 0; i < selectedProducts.length; i++) {
64             if (selectedProducts[i] == null) {
65                 selectedProducts[i] = (String) productsComboBox.getSelectedItem();
66                 JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
67                 break;
68             }
69         }
70     }
71 });
72 calculatePriceButton.addActionListener(new ActionListener() {
```



```

    }
});
calculatePriceButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        totalPrice = 0;
        for (String product : selectedProducts) {
            if (product != null) {
                if (product.contains(s: "Smartphone")) {
                    totalPrice += 599.9;
                } else if (product.contains(s: "T-shirt")) {
                    totalPrice += 19.99;
                } else if (product.contains(s: "OOP Book")) {
                    totalPrice += 39.99;
                }
            }
        }
        totalPriceLabel.setText("Total Price: $" + totalPrice);
    }
});
printOrderButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        StringBuilder orderInfo = new StringBuilder();
        orderInfo.append(str: "Customer ID: ").append(str: customerIdField.getText()).append(str: "\n");
        orderInfo.append(str: "Name: ").append(str: nameField.getText()).append(str: "\n");
        orderInfo.append(str: "Address: ").append(str: addressField.getText()).append(str: "\n");
        orderInfo.append(str: "Number of Products: ").append(str: nProductField.getText()).append(str: "\n");
        orderInfo.append(str: "Selected Products:\n");
        for (String product : selectedProducts) {
            if (product != null) {
                orderInfo.append(str: "- ").append(str: product).append(str: "\n");
            }
        }
        orderInfo.append(str: "Total Price: $").append(d: totalPrice);
    }
});

```

```

    ));
    printOrderButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            StringBuilder orderInfo = new StringBuilder();
            orderInfo.append(str: "Customer ID: ").append(str: customerIdField.getText()).append(str: "\n");
            orderInfo.append(str: "Name: ").append(str: nameField.getText()).append(str: "\n");
            orderInfo.append(str: "Address: ").append(str: addressField.getText()).append(str: "\n");
            orderInfo.append(str: "Number of Products: ").append(str: nProductField.getText()).append(str: "\n");
            orderInfo.append(str: "Selected Products:\n");
            for (String product : selectedProducts) {
                if (product != null) {
                    orderInfo.append(str: "- ").append(str: product).append(str: "\n");
                }
            }
            orderInfo.append(str: "Total Price: $").append(d: totalPrice);
            JOptionPane.showMessageDialog(parentComponent: null, message: orderInfo.toString(), title: "Order Information", messageType: JOptionPane.INFORMATION_MESSAGE);
        }
    });
    setVisible(b: true);
}
}

```

```
run:
Welcome to the E-Commerce System!
please enter your id
23010050
Please enter your name
Rokayya
Please enter your address
smouha
How many products you want to add to your cart?
4
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
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
1
The total is $679.8700256347656 Would you like to place the order? 1- Yes 2- No
1
OrderId: 1
CustomerId: 23010050
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
smartphone - $599.9
Total price: $679.87
BUILD SUCCESSFUL (total time: 18 seconds)
```