

```
Start Page X ECommerceSystem.java X Product.java X ElectronicProduct.java X ClothingProduct.java X BookProduct.java X Customer.java X Cart.java X Order.java X ECommerceGUI.java X
Source History
1 package ecommercesystem;
2 import java.util.Scanner;
3 public class ECommerceSystem {
4     public static void main(String[] args) {
5         ElectronicProduct electronicproduct = new ElectronicProduct(brand: "smartphone", warrantyPeriod:1, productId: 1, name: "Samsung", price: 599.99f);
6         ClothingProduct clothingproduct = new ClothingProduct(size: "Medium", fabric: "Cotton", productId: 2, name: "T-shirt", price: 19.99f);
7         BookProduct bookproduct = new BookProduct(author: "O'Reilly", publisher: "X Publications", productId: 3, name: "OOP", price: 39.99f );
8
9         Scanner cs=new Scanner(source: System.in);
10
11         System.out.println(x: "Welcome to the E-Commerce System!");
12         System.out.println(x: "Please enter your ID:");
13         int id=cs.nextInt();
14         cs.nextLine();
15
16         System.out.println(x: "Please enter your name:");
17         String name=cs.nextLine();
18
19         System.out.println(x: "Please enter your address:");
20         String address=cs.nextLine();
21
22         System.out.println(x: "How many products you want to add to your cart?");
23         int nProducts=cs.nextInt();
24         Product[] products = new Product[nProducts];
25
26         Customer customer = new Customer(customerId:id, name, address);
27         Cart cart = new Cart(customerId:id, nProducts, products);
28
29         for (int i=0; i<nProducts; i++) {
30             System.out.println(x: "Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
31             int choice = cs.nextInt();
32             cs.nextLine();
33             switch (choice) {
34                 case 1:
35                     cart.addProduct(pr: electronicproduct);
```

```

Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x
Source History
28
29 for (int i=0; i<nProducts; i++) {
30     System.out.println(x: "Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
31     int choice = cs.nextInt();
32     cs.nextLine();
33     switch (choice) {
34         case 1:
35             cart.addProduct(pr: electronicproduct);
36             break;
37         case 2:
38             cart.addProduct(pr: clothingproduct);
39             break;
40         case 3:
41             cart.addProduct(pr: bookproduct);
42             break;
43         default:
44             System.out.println(x: "Invalid choice!,choose from 1 to 3.");
45             i--;
46             continue;
47     }
48 }
49 float totalPrice = cart.calculatePrice();
50 System.out.println("Your total is $" +totalPrice+" Would you like to place the order? 1-Yes 2- No");
51 int choice=cs.nextInt();
52 if(choice==1){
53     Order order=new Order(customerId:customer.getId(),orderId:1,products: cart.getProducts(),totalPrice:cart.calculatePrice());
54     order.printOrderInfo();
55     cart.placeOrder();
56 }else
57     System.out.println(x: "Order wasn't placed");
58 System.out.println(x: "Would you like to remove a product from your cart? 1-Yes 2- No");
59
60 int removeChoice = cs.nextInt();
61 if (removeChoice == 1) {
62     System.out.println(x: "Enter the index of the product you want to remove:");

```

Output: ECommerceSystem (out) x

```
Start Page X ECommerceSystem.java X Product.java X ElectronicProduct.java X ClothingProduct.java X BookProduct.java X Customer.java X Cart.java X Order.java X ECommerceGUI.java X
Source History
46         continue;
47     }
48 }
49 float totalPrice = cart.calculatePrice();
50 System.out.println("Your total is $" + totalPrice + " Would you like to place the order? 1-Yes 2- No");
51 int choice = cs.nextInt();
52 if (choice == 1) {
53     Order order = new Order(customerId: customer.getCustomerId(), orderId: 1, products: cart.getProducts(), totalPrice: cart.calculatePrice());
54     order.printOrderInfo();
55     cart.placeOrder();
56 } else
57     System.out.println(x: "Order wasn't placed");
58 System.out.println(x: "Would you like to remove a product from your cart? 1-Yes 2- No");
59
60 int removeChoice = cs.nextInt();
61 if (removeChoice == 1) {
62     System.out.println(x: "Enter the index of the product you want to remove:");
63     int indexToRemove = cs.nextInt();
64     cart.removeProduct(index: indexToRemove);
65     System.out.println(x: "Product removed successfully!");
66 } else {
67     System.out.println(x: "No products were removed.");
68 }
69 Order order = new Order(customerId: customer.getCustomerId(), orderId: 1, products: cart.getProducts(), totalPrice: cart.calculatePrice());
70 order.printOrderInfo();
71 }
72 }
73 |
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x
Source History
1 package ecommercesystem;
2 public class Product {
3     protected int productId;
4     protected String name;
5     protected float price;
6
7     public Product(int productId, String name, float price) {
8         this.productId = Math.abs(productId);
9         this.name = name;
10        this.price = Math.abs(price);
11    }
12
13    public int getProductId() {
14        return productId;
15    }
16    public void setProductId(int productId) {
17        this.productId = Math.abs(productId);
18    }
19
20    public String getName() {
21        return name;
22    }
23    public void setName(String name) {
24        this.name = name;
25    }
26
27    public float getPrice() {
28        return price;
29    }
30    public void setPrice(float price) {
31        this.price = Math.abs(price);
32    }
33 }
```

```
Start Page X ECommerceSystem.java X Product.java X ElectronicProduct.java X ClothingProduct.java X BookProduct.java X Customer.java X Cart.java X Order.java X EComm
Source History
1 package ecommercesystem;
2 public class ElectronicProduct extends Product{
3     private String brand;
4     private int warrantyPeriod;
5
6     public ElectronicProduct(String brand, int warrantyPeriod, int productId, String name, float price) {
7         super(productId, name, price);
8         this.brand = brand;
9         this.warrantyPeriod = Math.abs(a: warrantyPeriod);
10    }
11
12    public String getBrand() {
13        return brand;
14    }
15    public void setBrand(String brand) {
16        this.brand = brand;
17    }
18
19    public int getWarrantyPeriod() {
20        return warrantyPeriod;
21    }
22    public void setWarrantyPeriod(int warrantyPeriod) {
23        this.warrantyPeriod = Math.abs(a: warrantyPeriod);
24    }
25 }
```

```
Start Page X ECommerceSystem.java X Product.java X ElectronicProduct.java X ClothingProduct.java X BookProduct.java X Customer.java X Cart.java X Order.java X ECommerceGUI.java X
Source History
1 package ecommercesystem;
2 public class ClothingProduct extends Product{
3     private String size;
4     private String fabric;
5
6     public ClothingProduct(String size, String fabric, int productId, String name, float price) {
7         super(productId, name, price);
8         this.size = size;
9         this.fabric = fabric;
10    }
11
12    public String getSize() {
13        return size;
14    }
15    public void setSize(String size) {
16        this.size = size;
17    }
18
19    public String getFabric() {
20        return fabric;
21    }
22    public void setFabric(String fabric) {
23        this.fabric = fabric;
24    }
25 }
26
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java
Source History
1 package ecommerceSystem;
2 public class BookProduct extends Product{
3     private String author;
4     private String publisher;
5
6     public BookProduct(String author, String publisher, int productId, String name, float price) {
7         super(productId, name, price);
8         this.author = author;
9         this.publisher = publisher;
10    }
11
12    public String getAuthor() {
13        return author;
14    }
15    public void setAuthor(String author) {
16        this.author = author;
17    }
18
19    public String getPublisher() {
20        return publisher;
21    }
22    public void setPublisher(String publisher) {
23        this.publisher = publisher;
24    }
25 }
```

Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.

Source History

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

```
Start Page × ECommerceSystem.java × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java
Source History
1 package ecommercesystem;
2 public class Cart {
3     private int customerId;
4     private int nProducts;
5     private Product[] products;
6
7     public Cart(int customerId, int nProducts, Product[] products) {
8         this.customerId = Math.abs(a: customerId);
9         this.nProducts = Math.abs(a: nProducts);
10        this.products = products;
11    }
12
13    public int getCustomerId() {
14        return customerId;
15    }
16
17    public void setCustomerId(int Id) {
18        if (Id>0){
19            customerId=Id;
20        }
21        else {
22            customerId= Math.abs(a: Id);
23        }
24    }
25    public int getnProducts() {
26        return nProducts;
27    }
28
29    public void setnProducts(int n) {
30        if (n>0){
31            nProducts=n;
32        }
33        else {
34            nProducts= Math.abs(a: n);
35        }
36    }
37
38    public Product[] getProducts() {
39        return products;
40    }
41
42    public void setProducts(Product[] products) {
43        this.products = products;
44    }
45
46    public void addProduct (Product pr){
47        for(int i =0 ; i<products.length ;i++){
48            if (products[i]==null){
49                products[i]=pr;
50                break;
51            }
52        }
53    }
54
55    public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
56        for (int i = index; i < nProducts - 1; i++) {
57            products[i] = products[i + 1];
58        }
59        products[nProducts - 1] = null;
60        nProducts--;
61    } else {
62        System.out.println(x: "Invalid index. Product not removed.");
63    }
64    }
```

```
Start Page × ECommerceSystem.java × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java
Source History
28
29    public void setnProducts(int n) {
30        if (n>0){
31            nProducts=n;
32        }
33        else {
34            nProducts= Math.abs(a: n);
35        }
36    }
37
38    public Product[] getProducts() {
39        return products;
40    }
41
42    public void setProducts(Product[] products) {
43        this.products = products;
44    }
45
46    public void addProduct (Product pr){
47        for(int i =0 ; i<products.length ;i++){
48            if (products[i]==null){
49                products[i]=pr;
50                break;
51            }
52        }
53    }
54
55    public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
56        for (int i = index; i < nProducts - 1; i++) {
57            products[i] = products[i + 1];
58        }
59        products[nProducts - 1] = null;
60        nProducts--;
61    } else {
62        System.out.println(x: "Invalid index. Product not removed.");
63    }
64    }
```



```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.ja
Source History
49         products[i]=pr;
50         break;
51     }
52 }
53 }
54 public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
55     for (int i = index; i < nProducts - 1; i++) {
56         products[i] = products[i + 1];
57     }
58     products[nProducts - 1] = null;
59     nProducts--;
60 }
61 } else {
62     System.out.println(x: "Invalid index. Product not removed.");
63 }
64 }
65 public float calculatePrice(){
66     float totalPrice=0;
67     for(int i =0; i<nProducts;i++){
68         totalPrice+= products[i].getPrice();
69     }
70     return totalPrice;
71 }
72
73 public void placeOrder(){
74     System.out.println(x: "your order is placed successfully!");
75 }
76 }
77
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x
Source History
1 package ecommercesystem;
2 public class Order {
3     private int customerId;
4     private int orderId;
5     private Product[] products;
6     private float totalPrice;
7
8     public Order(int customerId, int orderId, Product[] products, float totalPrice) {
9         this.customerId = Math.abs(a: customerId);
10        this.orderId = Math.abs(a: orderId);
11        this.products = products;
12        this.totalPrice = Math.abs(a: totalPrice);
13    }
14
15    public int getCustomerId() {
16        return customerId;
17    }
18
19    public void setCustomerId(int customerId) {
20        this.customerId = Math.abs(a: customerId);
21    }
22
23    public Product[] getProducts() {
24        return products;
25    }
26
27    public void setProducts(Product[] products) {
28        this.products = products;
29    }
30
31    public float getTotalPrice() {
32        return totalPrice;
33    }
34
35    public void setTotalPrice(float totalPrice) {
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
        public float getTotalPrice() {
            return totalPrice;
        }
        public void setTotalPrice(float totalPrice) {
            this.totalPrice = Math.abs(a: totalPrice);
        }
        public void setOrderId(int orderId) {
            this.orderId = Math.abs(a: orderId);
        }
        public int getOrderId() {
            return orderId;
        }
        public void printOrderInfo() {
            System.out.println(x: "Here's your order's summary:");
            System.out.println("Order's ID: "+orderId);
            System.out.println("Customer's ID: "+customerId);
            System.out.println(x: "Products: ");
            for (Product product : products) {
                if (product != null) {
                    System.out.println(product.getName() + " - $" + product.getPrice());
                }
            }
            System.out.println("Total price: $" + totalPrice);
        }
    }
}
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x
Source History
1 package ecommercesystem;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6
7 public class ECommerceGUI extends JFrame {
8     private JLabel customerIdLabel, nameLabel, addressLabel, productsLabel, totalPriceLabel;
9     private JTextField customerIdField, nameField, addressField;
10    private JComboBox<String> productsComboBox;
11    private JButton addProductButton, calculatePriceButton, printOrderButton;
12    private String[] selectedProducts = new String[3]; // Assuming a maximum of 3 products c
13    private double totalPrice = 0;
14
15    public ECommerceGUI() {
16        setTitle(title: "E-Commerce System");
17        setSize(width: 400, height: 300);
18        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
19        setLayout(new GridLayout(rows: 7, cols: 2));
20
21        customerIdLabel = new JLabel(text: "Customer ID:");
22        add(comp: customerIdLabel);
23        customerIdField = new JTextField();
24        add(comp: customerIdField);
25
26        nameLabel = new JLabel(text: "Name:");
27        add(comp: nameLabel);
28        nameField = new JTextField();
29        add(comp: nameField);
30
31        addressLabel = new JLabel(text: "Address:");
32        add(comp: addressLabel);
33        addressField = new JTextField();
34        add(comp: addressField);
35    }
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x ECommerceGUI.jav
Source History
35
36    productsLabel = new JLabel(text: "Select Product:");
37    add(comp: productsLabel);
38    String[] products = {"T-shirt - $19.99", "Smartphone - $599.9", "OOP Book - $39.99"};
39    productsComboBox = new JComboBox<>(items: products);
40    add(comp: productsComboBox);
41
42    addProductButton = new JButton(text: "Add Product");
43    add(comp: addProductButton);
44
45    totalPriceLabel = new JLabel(text: "Total Price: $0.0");
46    add(comp: totalPriceLabel);
47
48    calculatePriceButton = new JButton(text: "Calculate Total Price");
49    add(comp: calculatePriceButton);
50
51    printOrderButton = new JButton(text: "Print Order Info");
52    add(comp: printOrderButton);
53
54    addProductButton.addActionListener(new ActionListener() {
55        Override
56        public void actionPerformed(ActionEvent e) {
57            for (int i = 0; i < selectedProducts.length; i++) {
58                if (selectedProducts[i] == null) {
59                    selectedProducts[i] = (String) productsComboBox.getSelectedItem();
60                    JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
61                    break;
62                }
63            }
64        }
65    });
66
67    calculatePriceButton.addActionListener(new ActionListener() {
68        Override
69        public void actionPerformed(ActionEvent e) {
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x EComm

Source History

64      }
65      });
66
67      calculatePriceButton.addActionListener(new ActionListener() {
68          @Override
69          public void actionPerformed(ActionEvent e) {
70              totalPrice = 0;
71              for (String product : selectedProducts) {
72                  if (product != null) {
73                      if (product.contains(s: "T-shirt")) {
74                          totalPrice += 19.99;
75                      } else if (product.contains(s: "Smartphone")) {
76                          totalPrice += 599.9;
77                      } else if (product.contains(s: "OOP Book")) {
78                          totalPrice += 39.99;
79                      }
80                  }
81              }
82              totalPriceLabel.setText("Total Price: $" + totalPrice);
83          }
84      });
85
86      printOrderButton.addActionListener(new ActionListener() {
87          @Override
88          public void actionPerformed(ActionEvent e) {
89              StringBuilder orderInfo = new StringBuilder();
90              orderInfo.append(s: "Customer ID: ").append(s: customerIdField.getText()).append(s: "\n");
91              orderInfo.append(s: "Name: ").append(s: nameField.getText()).append(s: "\n");
92              orderInfo.append(s: "Address: ").append(s: addressField.getText()).append(s: "\n");
93              orderInfo.append(s: "Selected Products:\n");
94              for (String product : selectedProducts) {
95                  if (product != null) {
96                      orderInfo.append(s: "- ").append(s: product).append(s: "\n");
97                  }
98              }
99          }
100      });
101  }
```

```
Start Page x ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x

Source History

79      }
80      }
81      }
82      totalPriceLabel.setText("Total Price: $" + totalPrice);
83  }
84  });
85
86  printOrderButton.addActionListener(new ActionListener() {
87      @Override
88      public void actionPerformed(ActionEvent e) {
89          StringBuilder orderInfo = new StringBuilder();
90          orderInfo.append(s: "Customer ID: ").append(s: customerIdField.getText()).append(s: "\n");
91          orderInfo.append(s: "Name: ").append(s: nameField.getText()).append(s: "\n");
92          orderInfo.append(s: "Address: ").append(s: addressField.getText()).append(s: "\n");
93          orderInfo.append(s: "Selected Products:\n");
94          for (String product : selectedProducts) {
95              if (product != null) {
96                  orderInfo.append(s: "- ").append(s: product).append(s: "\n");
97              }
98          }
99          orderInfo.append(s: "Total Price: $").append(s: totalPrice);
100          JOptionPane.showMessageDialog(parentComponent: null, message: orderInfo.toString(), title: "Order Information", messageType: JOptionPane.INFORMATION_MESSAGE);
101      }
102  });
103
104  setVisible(s: true);
105  }
106
107  public static void main(String[] args) {
108      SwingUtilities.invokeLater(() -> new ECommerceGUI());
109  }
110  }
```

Output - ECommerceSystem (run) X

```
run:
Welcome to the E-Commerce System!
Please enter your ID:
23010085
Please enter your name:
Sarah Mohareb
Please enter your address:
smouha
How many products you want to add to your cart?
4
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
3
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
1
Your total is $679.95996 Would you like to place the order? 1-Yes 2- No
1
Here's your order's summary:
Order's ID: 1
Customer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Samsung - $599.99
Total price: $679.95996
your order is placed successfully!
Would you like to remove a product from your cart? 1-Yes 2- No
2
No products were removed.
Here's your order's summary:
Order's ID: 1
Customer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Samsung - $599.99
Total price: $679.95996
BUILD SUCCESSFUL (total time: 1 minute 15 seconds)
```

Output - ECommerceSystem (run) X



```
run:
Welcome to the E-Commerce System!
Please enter your ID:
23010085
Please enter your name:
Sarah Mohareb
Please enter your address:
smouha
How many products you want to add to your cart?

4
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
3
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
1
Your total is $679.95996 Would you like to place the order? 1-Yes 2- No
2
Order wasn't placed
Would you like to remove a product from your cart? 1-Yes 2- No
1
Enter the index of the product you want to remove:
2
Product removed successfully!
Here's your order's summary:
Order's ID: 1
Customer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
Samsung - $599.99
Total price: $659.97
BUILD SUCCESSFUL (total time: 52 seconds)
```

E-Commerce System

Customer ID: 23010085

Name: sarah

Address: smouha

Select Product: T-shirt - \$19.99

Add Product

Calculate Total Price

T-shirt - \$19.99

Smartphone - \$599.9

OOP Book - \$39.99

Print Order Info

E-Commerce System

Customer ID: 23010085

Name: sarah

Address: smouha

Select Product: Smartphone - \$599.9

Add Product

Calculate Total Price

Total Price: \$0.0

Message

Product added successfully!

OK

E-Commerce System

Customer ID:

23010085

Name:

sarah

Address:

smouha

Select Product:

Smartphone - \$599.9

Add Product

Total Price: \$1219.79

Calculate Total Price

Print Order Info

Order Information

i

Customer ID: 23010085

Name: sarah

Address: smouha

Selected Products:

- T-shirt - \$19.99

- Smartphone - \$599.9

- Smartphone - \$599.9

Total Price: \$1219.79

OK