```java
package com.mycompany.ecommercesystem;

public class Product {
    private int productId;
    private String name;
    private double price;

    public Product(int productId, String name, double price) {
        this.productId = Math.abs(a: productId);
        this.name = name;
        this.price = Math.abs(a: price);
    }
    public int getproductId(){
        return productId;
    }
    public void setproductId(int productId){
        this.productId = Math.abs(a: productId);
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = Math.abs(a: price);
    }
}
```

```java
package com.mycompany.ecommercesystem;

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

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

```java
package com.mycompany.ecommercesystem;

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

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

```java
package com.mycompany.ecommercesystem;

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

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

```java
package com.mycompany.ecommercesystem;

public class Customer {
    private int customerId;
    private String name;
    private String address;

    public Customer(int customerId, String name, String address) {
        this.customerId = Math.abs(a: customerId);
        this.name = name;
        this.address = address;
    }
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs(a: customerId);
    }
    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;
    }
}
```

```java
package com.mycompany.ecommercesystem;

public class Cart {
    private int customerId;
    private int nProducts;
    private Product[] products;

    public Cart(int customerId, int nProducts) {
        this.customerId = Math.abs(a: customerId);
        this.nProducts =  Math.abs(a: nProducts);
        this.products = new Product[nProducts];
    }
    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs(a: customerId);
    }
    public int getnProducts() {
        return nProducts;
    }
    public void setnProducts(int nProducts) {
        this.nProducts = Math.abs(a: nProducts);
    }
    public Product[] getProducts() {
        return products;
    }
    public void setProducts(Product[] products) {
        this.products = products;
    }
    public void addProducts(Product product , int index){
        if (index >= 0 && index < nProducts){
            products[index]=product;
        }else{
            System.out.println(x: "invalid index.");
        }
```

```
    }
    public Product[] getProducts() {
        return products;
    }
    public void setProducts(Product[] products) {
        this.products = products;
    }
    public void addProducts(Product product , int index){
        if (index >= 0 && index < nProducts){
            products[index]=product;
        }else{
            System.out.println(x: "invalid index.");
        }
    }
    public void removeProducts(int index){
     if (index >= 0 && index < nProducts) {
            products[index] = null;
        } else {
            System.out.println(x: "Invalid index.");
        }
    }
     public double calculatePrice() {
        double totalPrice = 0;
        for (Product product : products) {
                totalPrice += product.getPrice();
        }
        return totalPrice;
    }
      public void placeOrder() {
          System.out.println(x: "order placed successfully!");
      }
```

```
package com.mycompany.ecommercesystem;

public class Order {
    private int customerId;
    private int orderId;
    private Product[] products;
    private double totalPrice;

    public Order(int customerId, int orderId, Product[] products, double totalPrice) {
        this.customerId = Math.abs(a: customerId);
        this.orderId = Math.abs(a: orderId);
        this.products = products;
        this.totalPrice = Math.abs(a: totalPrice);
    }
    public int getCustomerId() {
        return customerId;
    }

    public void setCustomerId(int customerId) {
        this.customerId = Math.abs(a: customerId);
    }

    public int getOrderId() {
        return orderId;
    }

    public void setOrderId(int orderId) {
        this.orderId = Math.abs(a: orderId);
    }

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

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

```java
    public int getOrderId() {
        return orderId;
    }

    public void setOrderId(int orderId) {
        this.orderId = Math.abs(a: orderId);
    }

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

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

    public double getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(double totalPrice) {
        this.totalPrice = Math.abs(a: 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 (Product product : products) {
            System.out.println(product.getName() + " - $" + product.getPrice());
        }
        System.out.println("Total Price: $" + totalPrice);
    }
    }
```

```java
package com.mycompany.ecommercesystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class GUI extends JFrame {
    private JLabel customerIdLabel, nameLabel, addressLabel, productsLabel, totalPriceLabel;
    private JTextField customerIdField, nameField, addressField;
    private JComboBox<String> productsComboBox;
    private JButton addProductButton, calculatePriceButton, printOrderButton;
    private String[] selectedProducts = new String[3]; // Assuming a maximum of 3 products can be selected
    private double totalPrice = 0;

    public GUI() {
        setTitle(title: "E-Commerce System");
        setSize(width: 400, height:300);
        setDefaultCloseOperation(operation:JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(rows: 7, cols: 2));

        customerIdLabel = new JLabel(text: "Customer ID:");
        add(comp: customerIdLabel);
        customerIdField = new JTextField();
        add(comp: customerIdField);

        nameLabel = new JLabel(text: "Name:");
        add(comp: nameLabel);
        nameField = new JTextField();
        add(comp: nameField);

        addressLabel = new JLabel(text: "Address:");
        add(comp: addressLabel);
        addressField = new JTextField();
        add(comp: addressField);

        productsLabel = new JLabel(text: "Select Product:");
        add(comp: productsLabel);
```

```
nameLabel = new JLabel(text: "Name:");
add(comp: nameLabel);
nameField = new JTextField();
add(comp: nameField);

addressLabel = new JLabel(text: "Address:");
add(comp: addressLabel);
addressField = new JTextField();
add(comp: addressField);

productsLabel = new JLabel(text: "Select Product:");
add(comp: productsLabel);
String[] products = {"T-shirt - $19.99", "Smartphone - $599.9", "OOP Book - $39.99"};
productsComboBox = new JComboBox<>(items: products);
add(comp: productsComboBox);

addProductButton = new JButton(text: "Add Product");
add(comp: addProductButton);

totalPriceLabel = new JLabel(text: "Total Price: $0.0");
add(comp: totalPriceLabel);

calculatePriceButton = new JButton(text: "Calculate Total Price");
add(comp: calculatePriceButton);

printOrderButton = new JButton(text: "Print Order Info");
add(comp: printOrderButton);

addProductButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        for (int i = 0; i < selectedProducts.length; i++) {
            if (selectedProducts[i] == null) {
                selectedProducts[i] = (String) productsComboBox.getSelectedItem();
                JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
                break;
```

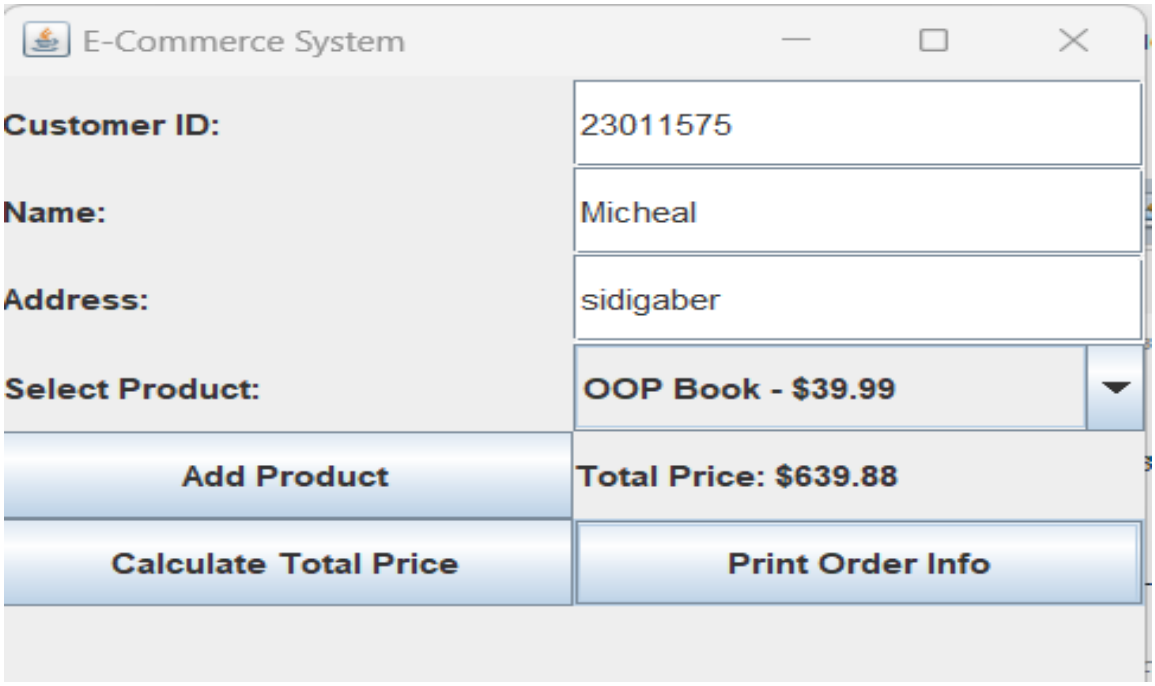
```
calculatePriceButton = new JButton(text: "Calculate Total Price");
add(comp: calculatePriceButton);

printOrderButton = new JButton(text: "Print Order Info");
add(comp: printOrderButton);

addProductButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        for (int i = 0; i < selectedProducts.length; i++) {
            if (selectedProducts[i] == null) {
                selectedProducts[i] = (String) productsComboBox.getSelectedItem();
                JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
                break;
            }
        }
    }
});

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

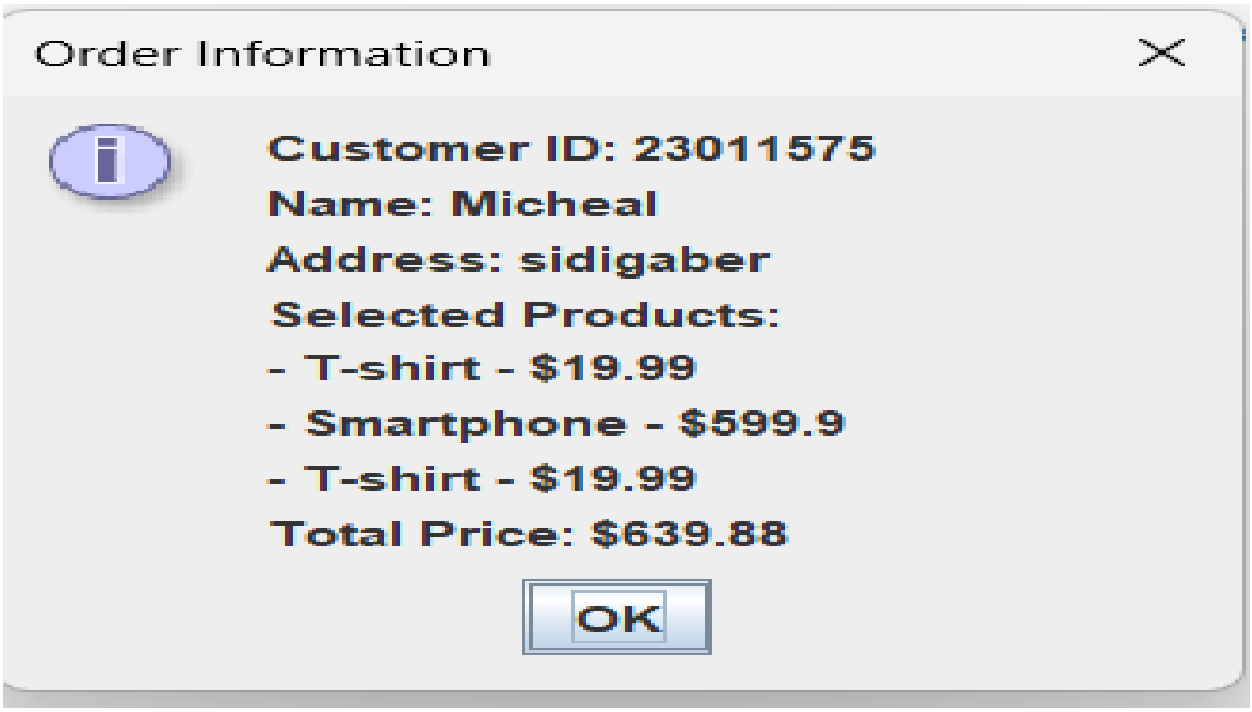
```
                } else if (product.contains("OOP Book")) {
                    totalPrice += 39.99;
                }
            }
        }
        totalPriceLabel.setText("Total Price: $" + totalPrice);
    }
});

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: "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);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new GUI());
}
}
```

```
...3 lines
package com.mycompany.ecommercesystem;
import java.util.Scanner;

public class EcommerceSystem {

    public static void main(String[] args) {
    Scanner input = new Scanner(source:System.in);

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

        System.out.println(x: "Welcome to E-commerce system|");
        System.out.println(x: "Please enter your id ");
        int Id = input.nextInt();
        System.out.println(x: "Please enter your name ");
        String name = input.next();
        System.out.println(x: "Please enter your address ");
        String address = input.next();

        Customer customer = new Customer(customerId: Id,name,address);
        System.out.println(x: "How many products you want to add to your cart?");
        int nProducts = input.nextInt();
        Cart cart = new Cart(customerId: customer.getCustomerId(), nProducts);

        for(int i=0; i<nProducts;i++){
            System.out.println(x: "Which product would you like to add? 1-Smartphone 2-T-shirt 3-OOP");
            int select = input.nextInt();
            Product product;
            switch (select) {
                case 1:
                    product = electronicProduct;
                    break;
                case 2:
                    product = clothingProduct;
                    break;
```

```
                product = electronicProduct;
                break;
            case 2:
                product = clothingProduct;
                break;
            case 3:
                product = bookProduct;
                break;
            default:
                System.out.println(x: "Invalid choice,You should choose from 1 to 3! ");
                i--;
                continue;
        }
        cart.addProducts(product, index: i);
    }
        System.out.print("Your total is $" + cart.calculatePrice()+". Do you want to place the order? 1-Yes 2-No: ");
    int choice = input.nextInt();
    if (choice == 1) {
       Order order = new Order(customerId: customer.getCustomerId(), orderId: 1, products: cart.getProducts(),totalPrice: cart.calculatePrice());
        order.printOrderInfo();
        cart.placeOrder();
    } else {
        System.out.println(x: "Order not placed.");
    }
    System.out.println(x: "would you like to remove a product from your cart? 1-Yes 2-No");
      int choiceRemove = input.nextInt();
      if (choiceRemove == 1) {
          System.out.println(x: "Enter the index of the product you want to remove:");
            int indexToremove = input.nextInt();
        cart.removeProducts(index: indexToremove);
          System.out.println(x: "Product removed. ");
      }else{
          System.out.println(x: "No products were removed.");
      }
    }
```

```
Welcome to E-commerce system|
Please enter your id
23011575
Please enter your name
Micheal
Please enter your address
sidigaber
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
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
Which product would you like to add? 1-Smartphone 2-T-shirt 3-OOP
2
Your total is $679.87. Do you want to place the order? 1-Yes 2-No: 1
Here's your order's summary:
Order ID: 1
Customer ID: 23011575
Products:
smartphone - $599.9
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Total Price: $679.87
order placed successfully!
would you like to remove a product from your cart? 1-Yes 2-No
2
No products were removed.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  31.153 s
Finished at: 2024-04-22T22:05:15+02:00
------------------------------------------------------------------------
```


E-Commerce System
Customer ID:
23011575
Name:
Micheal
Address:
sidigaber
Select Product:
OOP Book - $39.99
Add Product
Total Price: $639.88
Calculate Total Price
Print Order Info


Order Information
Customer ID: 23011575
Name: Micheal
Address: sidigaber
Selected Products:
- T-shirt - $19.99
- Smartphone - $599.9
- T-shirt - $19.99
Total Price: $639.88
OK