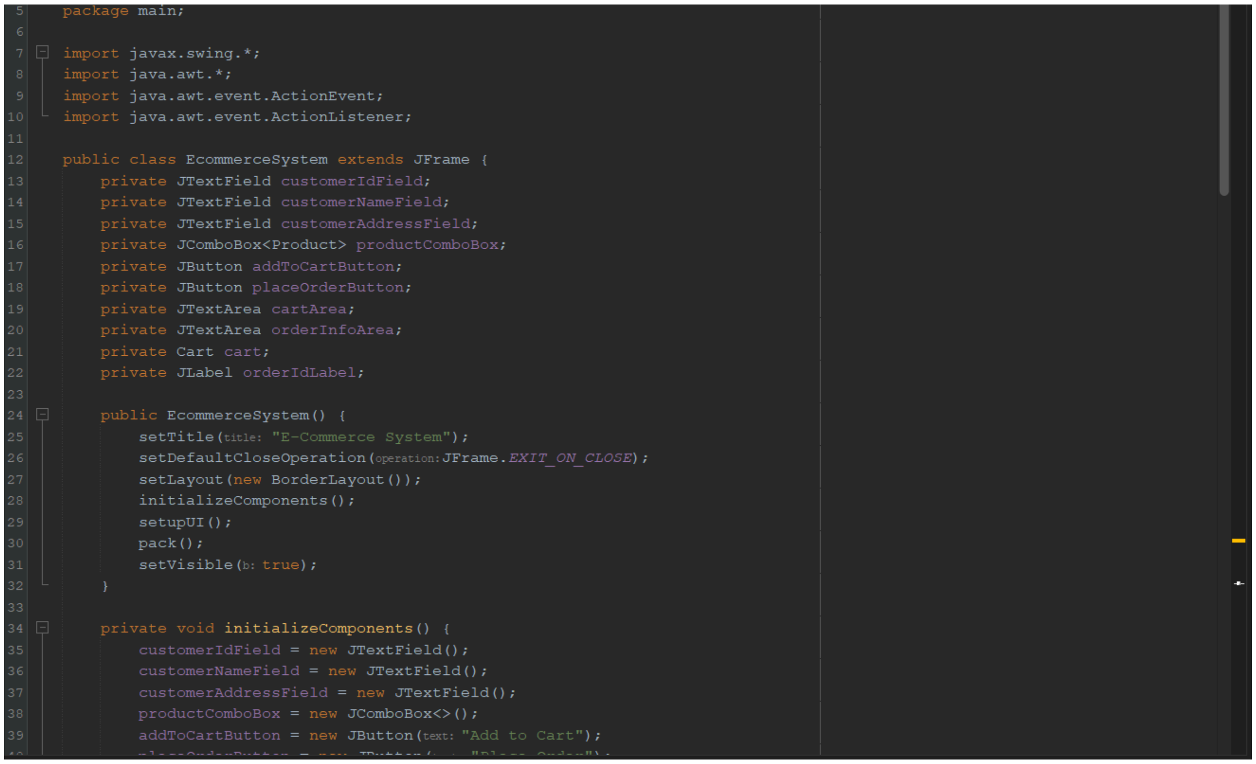
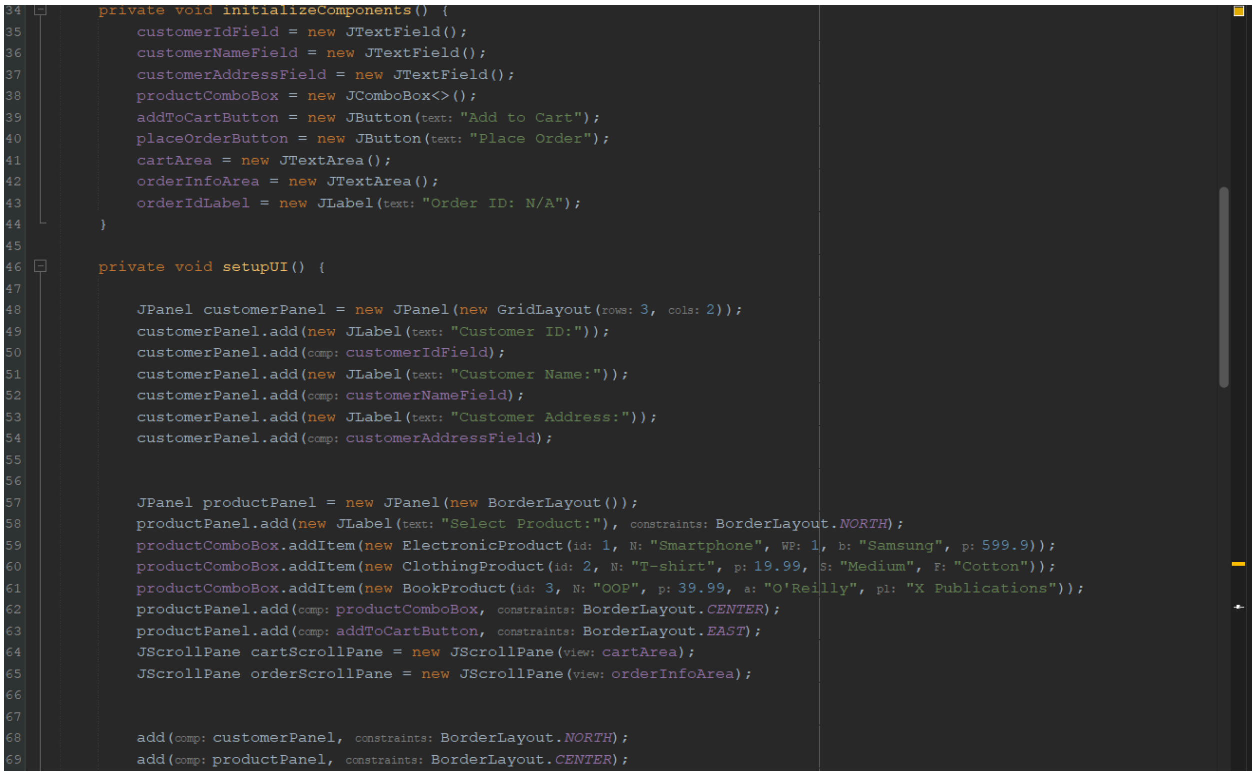
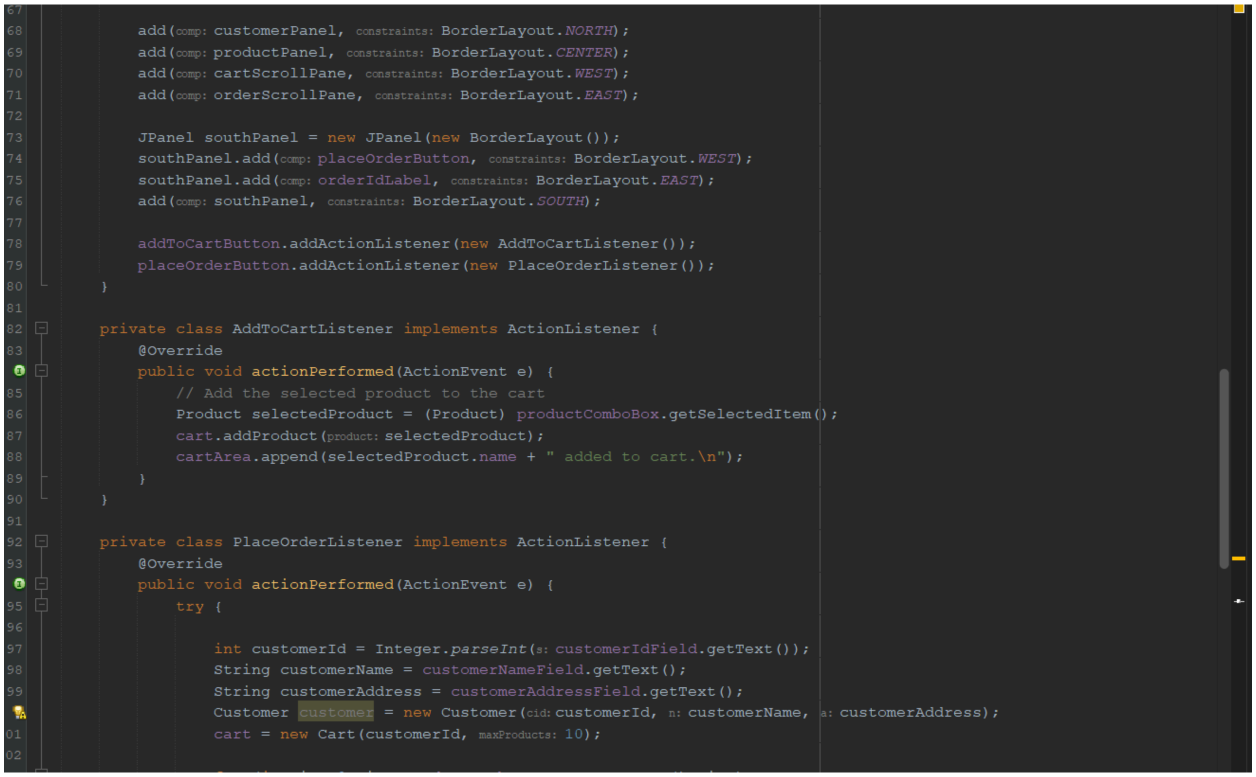
The Main:



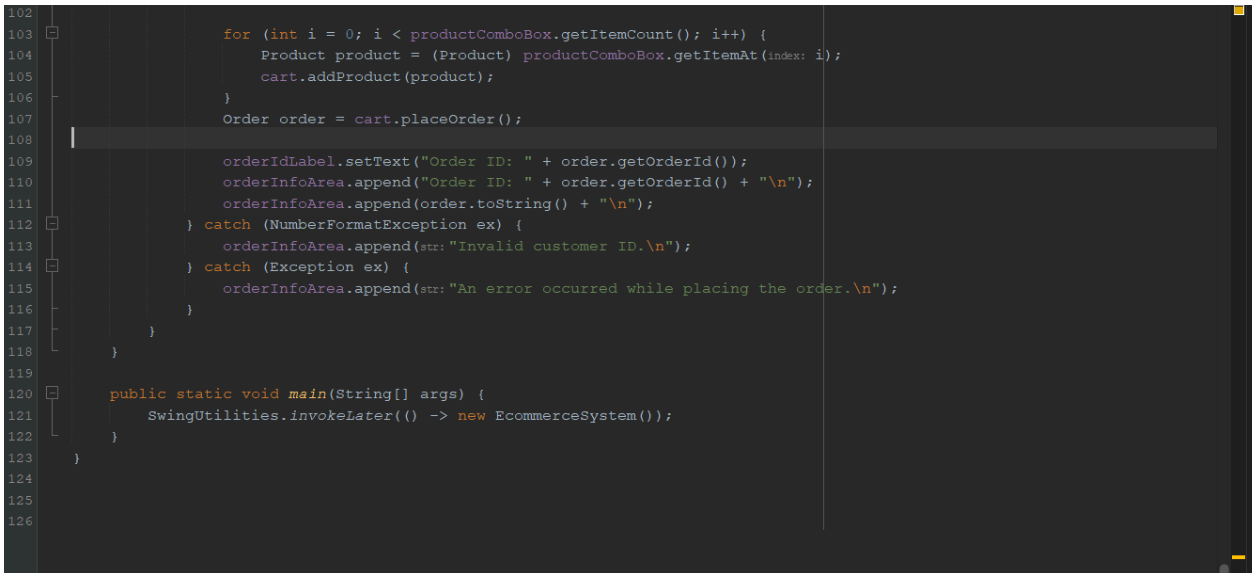
```java
package main;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class EcommerceSystem extends JFrame {
    private JTextField customerIdField;
    private JTextField customerNameField;
    private JTextField customerAddressField;
    private JComboBox<Product> productComboBox;
    private JButton addToCartButton;
    private JButton placeOrderButton;
    private JTextArea cartArea;
    private JTextArea orderInfoArea;
    private Cart cart;
    private JLabel orderIdLabel;

    public EcommerceSystem() {
        setTitle(title: "E-Commerce System");
        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        initializeComponents();
        setupUI();
        pack();
        setVisible(b: true);
    }

    private void initializeComponents() {
        customerIdField = new JTextField();
        customerNameField = new JTextField();
        customerAddressField = new JTextField();
        productComboBox = new JComboBox<>();
        addToCartButton = new JButton(text: "Add to Cart");
```



```java
    private void initializeComponents() {
        customerIdField = new JTextField();
        customerNameField = new JTextField();
        customerAddressField = new JTextField();
        productComboBox = new JComboBox<>();
        addToCartButton = new JButton(text: "Add to Cart");
        placeOrderButton = new JButton(text: "Place Order");
        cartArea = new JTextArea();
        orderInfoArea = new JTextArea();
        orderIdLabel = new JLabel(text: "Order ID: N/A");
    }

    private void setupUI() {

        JPanel customerPanel = new JPanel(new GridLayout(rows: 3, cols: 2));
        customerPanel.add(new JLabel(text: "Customer ID:"));
        customerPanel.add(comp: customerIdField);
        customerPanel.add(new JLabel(text: "Customer Name:"));
        customerPanel.add(comp: customerNameField);
        customerPanel.add(new JLabel(text: "Customer Address:"));
        customerPanel.add(comp: customerAddressField);


        JPanel productPanel = new JPanel(new BorderLayout());
        productPanel.add(new JLabel(text: "Select Product:"), constraints: BorderLayout.NORTH);
        productComboBox.addItem(new ElectronicProduct(id: 1, N: "Smartphone", WP: 1, b: "Samsung", p: 599.9));
        productComboBox.addItem(new ClothingProduct(id: 2, N: "T-shirt", p: 19.99, S: "Medium", F: "Cotton"));
        productComboBox.addItem(new BookProduct(id: 3, N: "OOP", p: 39.99, a: "O'Reilly", pl: "X Publications"));
        productPanel.add(comp: productComboBox, constraints: BorderLayout.CENTER);
        productPanel.add(comp: addToCartButton, constraints: BorderLayout.EAST);
        JScrollPane cartScrollPane = new JScrollPane(view: cartArea);
        JScrollPane orderScrollPane = new JScrollPane(view: orderInfoArea);


        add(comp: customerPanel, constraints: BorderLayout.NORTH);
        add(comp: productPanel, constraints: BorderLayout.CENTER);
```

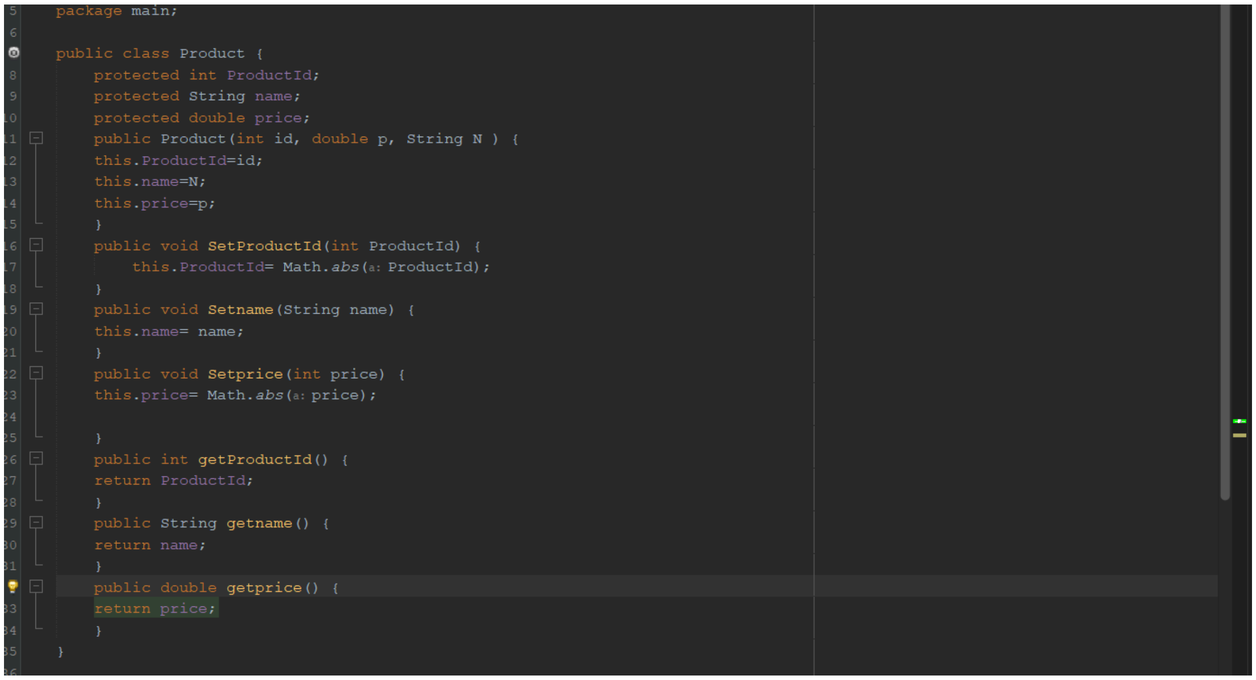


```java

        add(comp: customerPanel, constraints: BorderLayout.NORTH);
        add(comp: productPanel, constraints: BorderLayout.CENTER);
        add(comp: cartScrollPane, constraints: BorderLayout.WEST);
        add(comp: orderScrollPane, constraints: BorderLayout.EAST);

        JPanel southPanel = new JPanel(new BorderLayout());
        southPanel.add(comp: placeOrderButton, constraints: BorderLayout.WEST);
        southPanel.add(comp: orderIdLabel, constraints: BorderLayout.EAST);
        add(comp: southPanel, constraints: BorderLayout.SOUTH);

        addToCartButton.addActionListener(new AddToCartListener());
        placeOrderButton.addActionListener(new PlaceOrderListener());
    }

    private class AddToCartListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Add the selected product to the cart
            Product selectedProduct = (Product) productComboBox.getSelectedItem();
            cart.addProduct(product: selectedProduct);
            cartArea.append(selectedProduct.name + " added to cart.\n");
        }
    }

    private class PlaceOrderListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {

                int customerId = Integer.parseInt(s: customerIdField.getText());
                String customerName = customerNameField.getText();
                String customerAddress = customerAddressField.getText();
                Customer customer = new Customer(cid: customerId, n: customerName, a: customerAddress);
                cart = new Cart(customerId, maxProducts: 10);

```



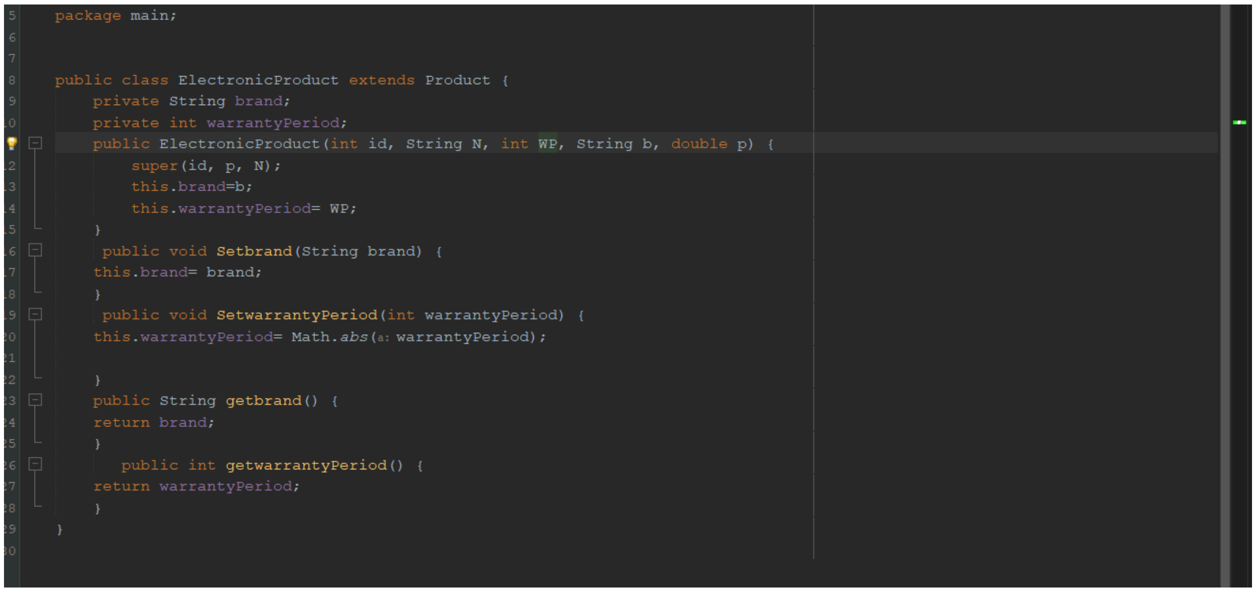
```java

                for (int i = 0; i < productComboBox.getItemCount(); i++) {
                    Product product = (Product) productComboBox.getItemAt(index: i);
                    cart.addProduct(product);
                }
                Order order = cart.placeOrder();

                orderIdLabel.setText("Order ID: " + order.getOrderId());
                orderInfoArea.append("Order ID: " + order.getOrderId() + "\n");
                orderInfoArea.append(order.toString() + "\n");
            } catch (NumberFormatException ex) {
                orderInfoArea.append(str: "Invalid customer ID.\n");
            } catch (Exception ex) {
                orderInfoArea.append(str: "An error occurred while placing the order.\n");
            }
        }
    }

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



```

Product Class:



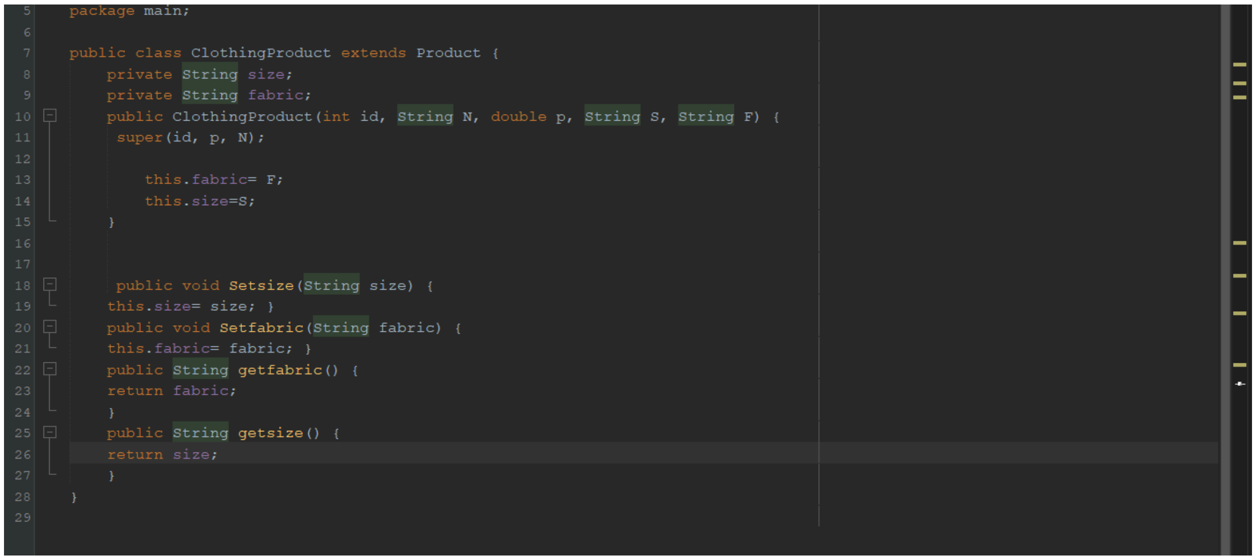
```java
package main;

public class Product {
    protected int ProductId;
    protected String name;
    protected double price;
    public Product(int id, double p, String N ) {
    this.ProductId=id;
    this.name=N;
    this.price=p;
    }
    public void SetProductId(int ProductId) {
        this.ProductId= Math.abs(a: ProductId);
    }
    public void Setname(String name) {
    this.name= name;
    }
    public void Setprice(int price) {
    this.price= Math.abs(a: price);

    }
    public int getProductId() {
    return ProductId;
    }
    public String getname() {
    return name;
    }
    public double getprice() {
    return price;
    }
}
```

Electronic Product Class:



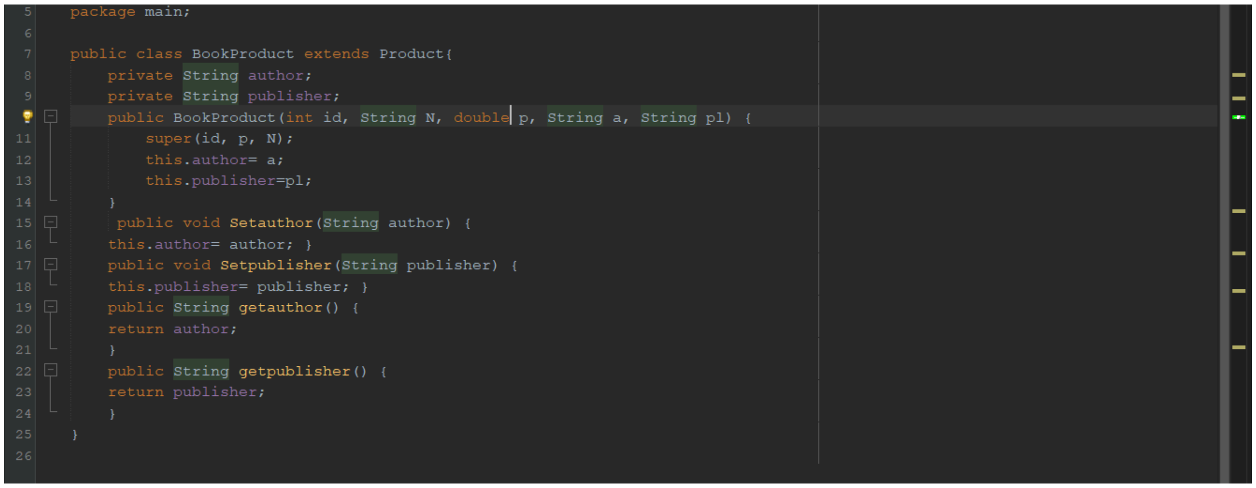
```java
package main;


public class ElectronicProduct extends Product {
    private String brand;
    private int warrantyPeriod;
    public ElectronicProduct(int id, String N, int WP, String b, double p) {
        super(id, p, N);
        this.brand=b;
        this.warrantyPeriod= WP;
    }
     public void Setbrand(String brand) {
    this.brand= brand;
    }
     public void SetwarrantyPeriod(int warrantyPeriod) {
    this.warrantyPeriod= Math.abs(a: warrantyPeriod);

    }
    public String getbrand() {
    return brand;
    }
       public int getwarrantyPeriod() {
    return warrantyPeriod;
    }
}
```

Clothing Product Class:



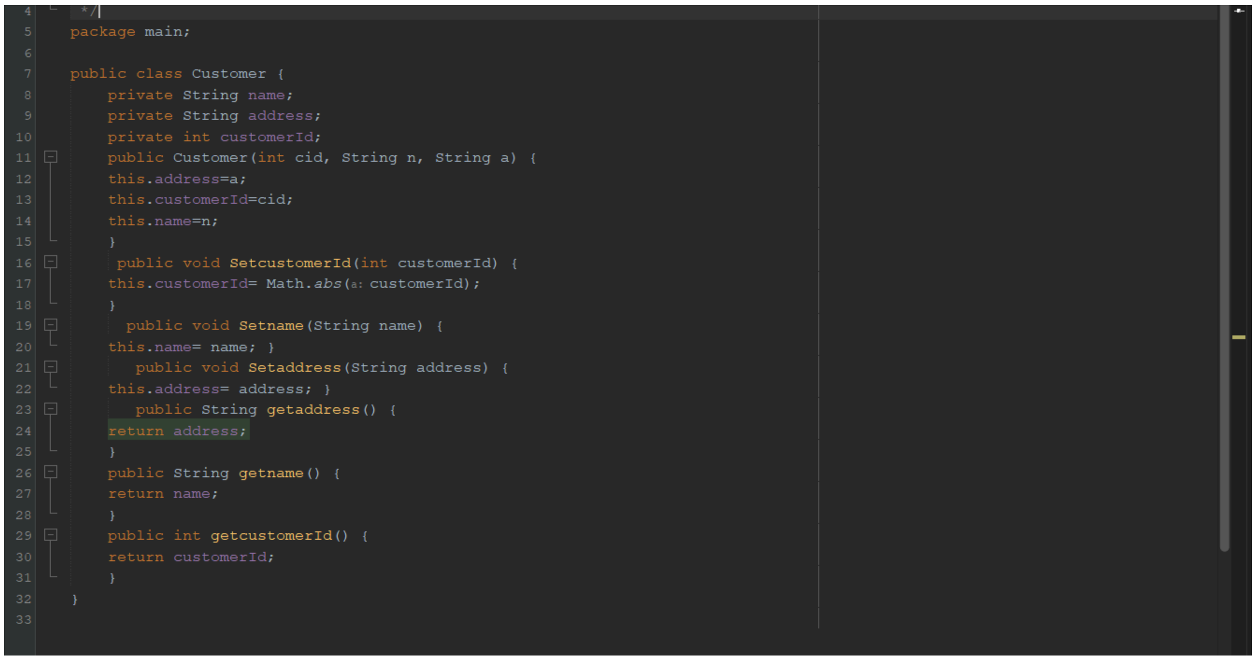
```
package main;

public class ClothingProduct extends Product {
    private String size;
    private String fabric;
    public ClothingProduct(int id, String N, double p, String S, String F) {
     super(id, p, N);

        this.fabric= F;
        this.size=S;
    }


     public void Setsize(String size) {
    this.size= size; }
    public void Setfabric(String fabric) {
    this.fabric= fabric; }
    public String getfabric() {
    return fabric;
    }
    public String getsize() {
    return size;
    }
}

```

Book Product Class:



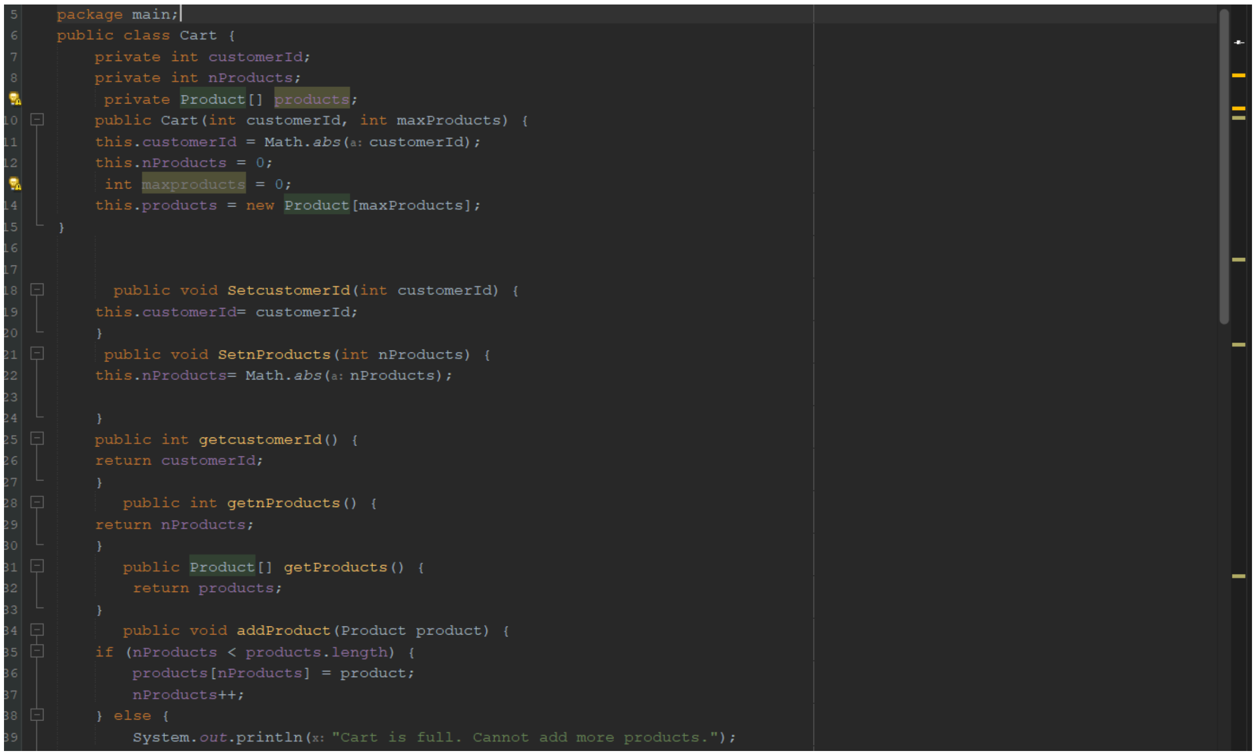
```
package main;

public class BookProduct extends Product{
    private String author;
    private String publisher;
    public BookProduct(int id, String N, double p, String a, String pl) {
        super(id, p, N);
        this.author= a;
        this.publisher=pl;
    }
     public void Setauthor(String author) {
    this.author= author; }
    public void Setpublisher(String publisher) {
    this.publisher= publisher; }
    public String getauthor() {
    return author;
    }
    public String getpublisher() {
    return publisher;
    }
}

```

Customer Class:

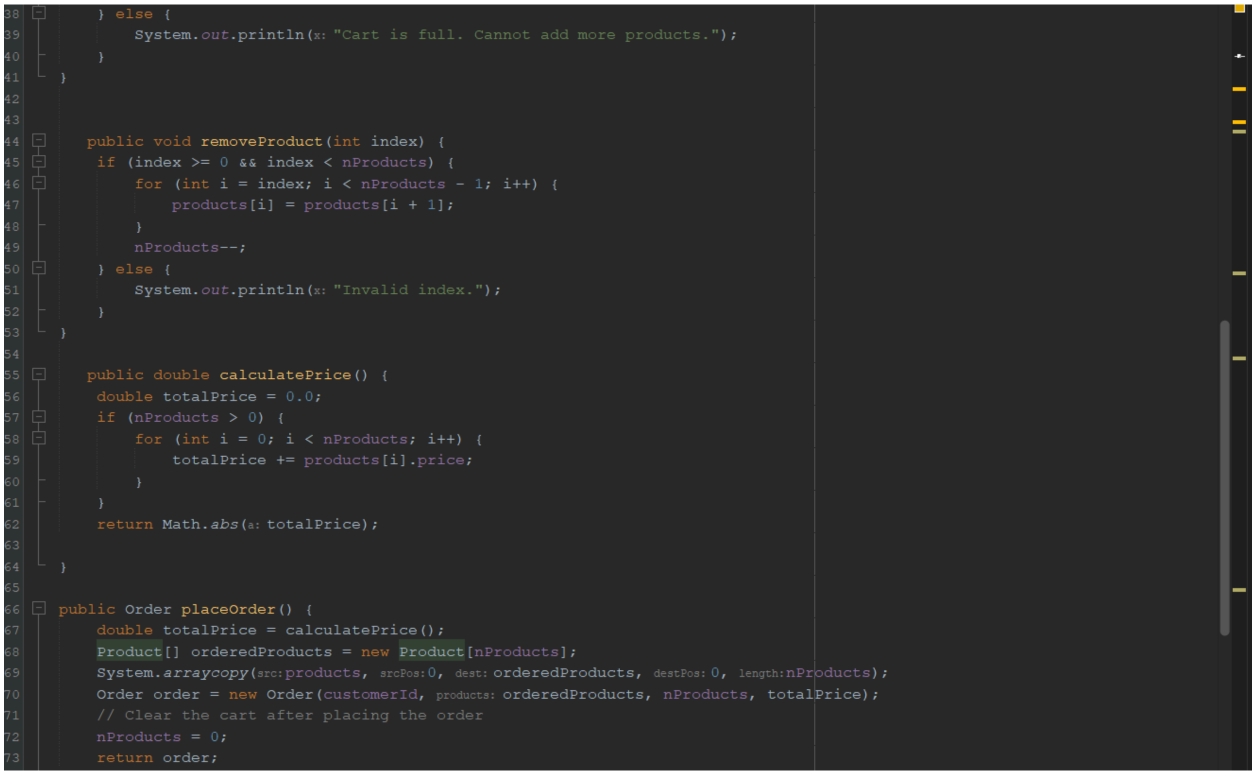


```
  */
package main;

public class Customer {
    private String name;
    private String address;
    private int customerId;
    public Customer(int cid, String n, String a) {
    this.address=a;
    this.customerId=cid;
    this.name=n;
    }
     public void SetcustomerId(int customerId) {
    this.customerId= Math.abs(a: customerId);
    }
      public void Setname(String name) {
    this.name= name; }
       public void Setaddress(String address) {
    this.address= address; }
       public String getaddress() {
    return address;
    }
    public String getname() {
    return name;
    }
    public int getcustomerId() {
    return customerId;
    }
}
```

Cart Class:

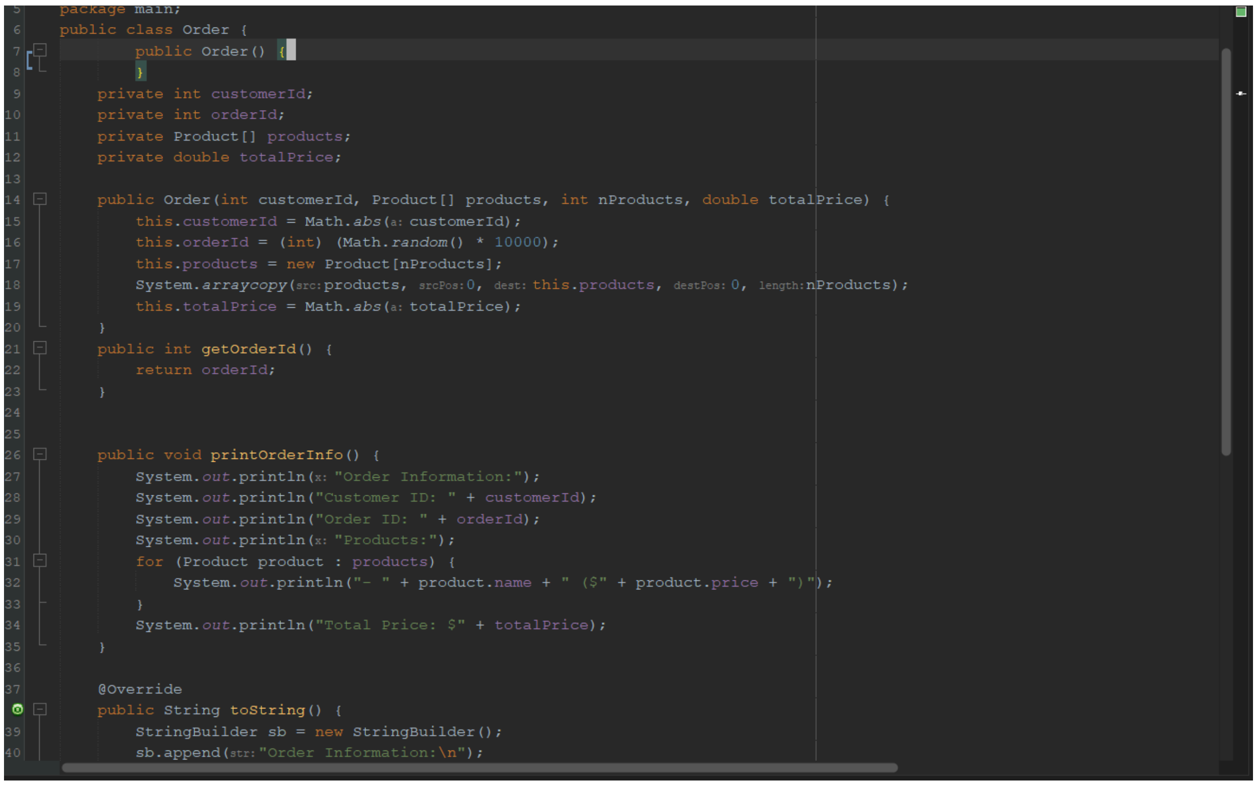


```
package main;
public class Cart {
    private int customerId;
    private int nProducts;
     private Product[] products;
    public Cart(int customerId, int maxProducts) {
    this.customerId = Math.abs(a: customerId);
    this.nProducts = 0;
     int maxproducts = 0;
    this.products = new Product[maxProducts];
}


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

    }
    public int getcustomerId() {
    return customerId;
    }
       public int getnProducts() {
    return nProducts;
    }
       public Product[] getProducts() {
        return products;
    }
       public void addProduct(Product product) {
    if (nProducts < products.length) {
        products[nProducts] = product;
        nProducts++;
    } else {
        System.out.println(x: "Cart is full. Cannot add more products.");
```

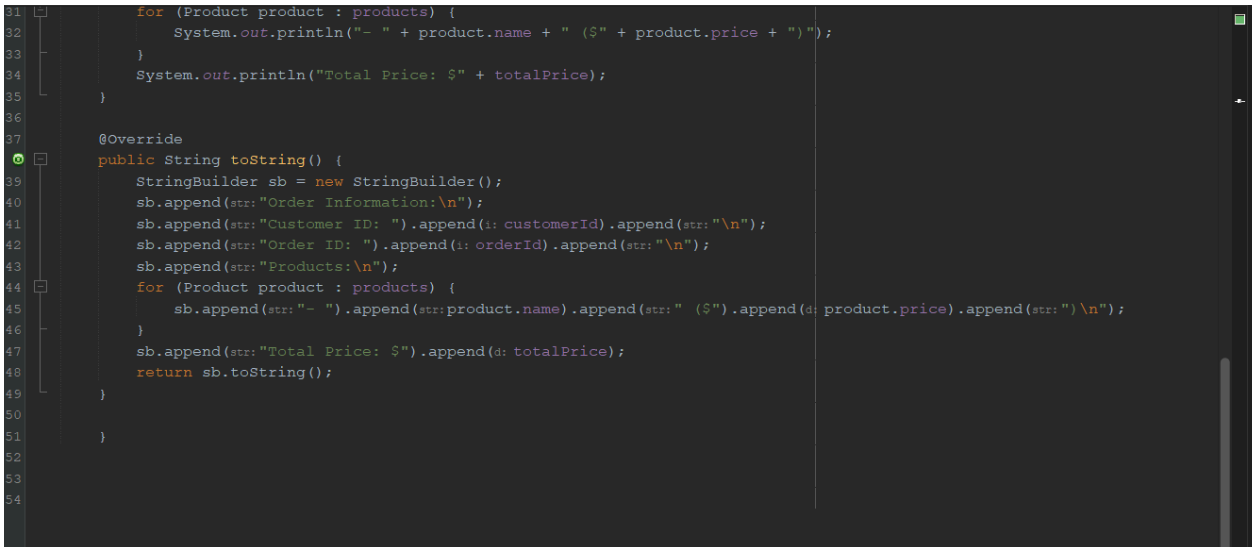


```
        } else {
            System.out.println(x: "Cart is full. Cannot add more products.");
        }
    }


       public void removeProduct(int index) {
        if (index >= 0 && index < nProducts) {
            for (int i = index; i < nProducts - 1; i++) {
                products[i] = products[i + 1];
            }
            nProducts--;
        } else {
            System.out.println(x: "Invalid index.");
        }
    }

       public double calculatePrice() {
        double totalPrice = 0.0;
        if (nProducts > 0) {
            for (int i = 0; i < nProducts; i++) {
                totalPrice += products[i].price;
            }
        }
        return Math.abs(a: totalPrice);

    }

    public Order placeOrder() {
        double totalPrice = calculatePrice();
        Product[] orderedProducts = new Product[nProducts];
        System.arraycopy(src:products, srcPos:0, dest: orderedProducts, destPos: 0, length:nProducts);
        Order order = new Order(customerId, products: orderedProducts, nProducts, totalPrice);
        // Clear the cart after placing the order
        nProducts = 0;
        return order;
```

Order Class:

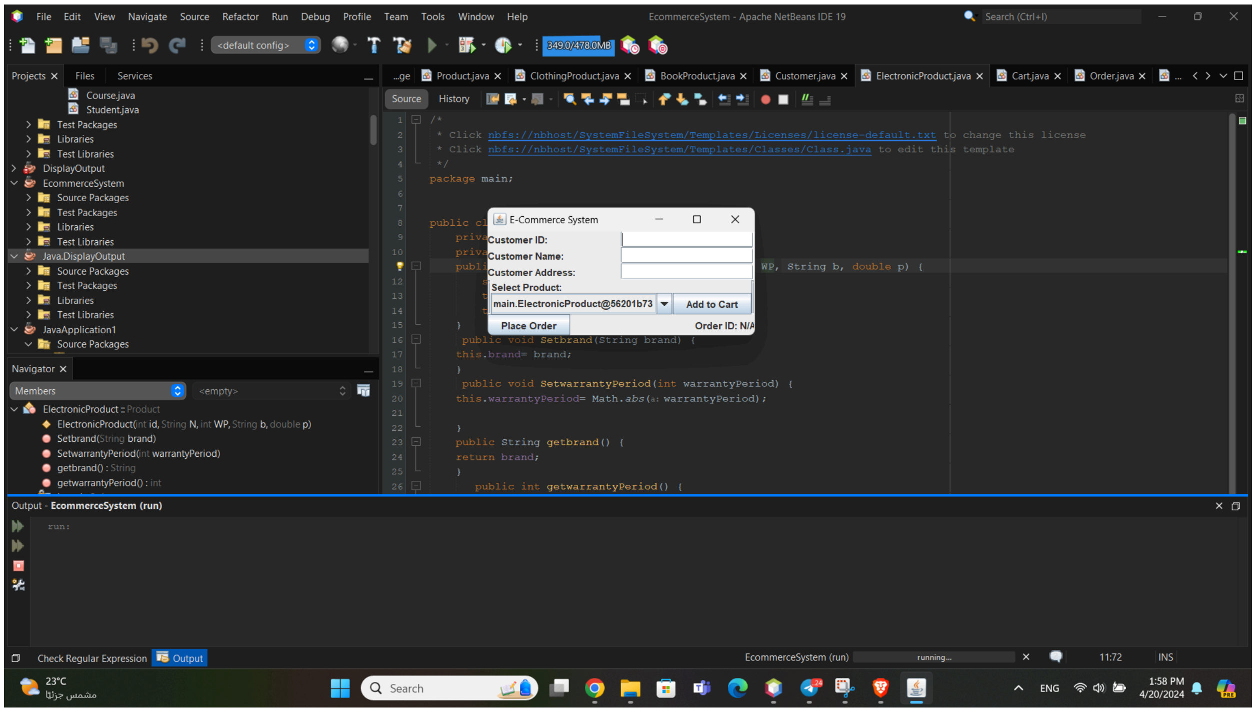


```
    package main;
    public class Order {
            public Order() {
            }
        private int customerId;
        private int orderId;
        private Product[] products;
        private double totalPrice;

        public Order(int customerId, Product[] products, int nProducts, double totalPrice) {
            this.customerId = Math.abs(a: customerId);
            this.orderId = (int) (Math.random() * 10000);
            this.products = new Product[nProducts];
            System.arraycopy(src:products, srcPos:0, dest: this.products, destPos: 0, length:nProducts);
            this.totalPrice = Math.abs(a: totalPrice);
        }
        public int getOrderId() {
            return orderId;
        }


        public void printOrderInfo() {
            System.out.println(x: "Order Information:");
            System.out.println("Customer ID: " + customerId);
            System.out.println("Order ID: " + orderId);
            System.out.println(x: "Products:");
            for (Product product : products) {
                System.out.println("- " + product.name + " ($" + product.price + ")");
            }
            System.out.println("Total Price: $" + totalPrice);
        }

        @Override
        public String toString() {
            StringBuilder sb = new StringBuilder();
            sb.append(str:"Order Information:\n");
```



```
        for (Product product : products) {
            System.out.println("- " + product.name + " ($" + product.price + ")");
        }
        System.out.println("Total Price: $" + totalPrice);
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append(str:"Order Information:\n");
        sb.append(str:"Customer ID: ").append(i: customerId).append(str:"\n");
        sb.append(str:"Order ID: ").append(i: orderId).append(str:"\n");
        sb.append(str:"Products:\n");
        for (Product product : products) {
            sb.append(str:"- ").append(str:product.name).append(str:" ($").append(d: product.price).append(str:")\n");
        }
        sb.append(str:"Total Price: $").append(d: totalPrice);
        return sb.toString();
    }

    }
```

Running Test:



EcommerceSystem - Apache NetBeans IDE 19

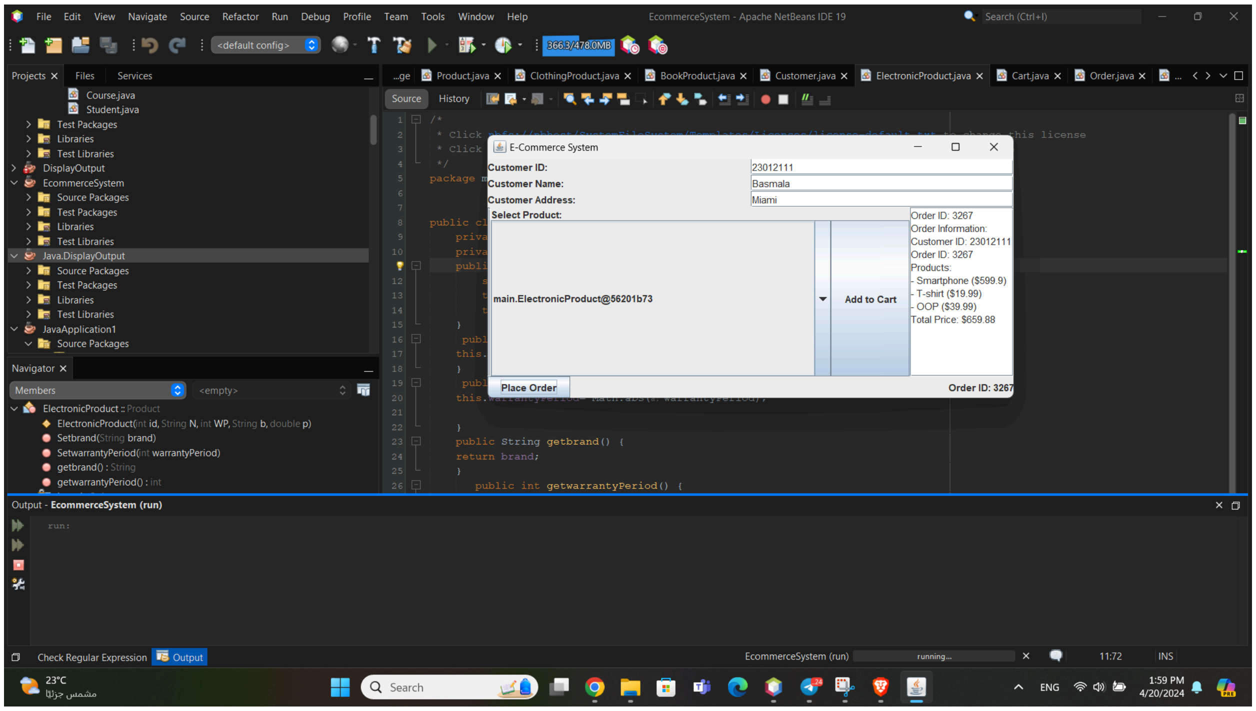
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package main;


public cl
    priva
    priva
    publi                                         WP, String b, double p) {
        s
        t
        t
    }
     public void Setbrand(String brand) {
    this.brand= brand;
    }
     public void SetwarrantyPeriod(int warrantyPeriod) {
    this.warrantyPeriod= Math.abs(a: warrantyPeriod);

    }
    public String getbrand() {
    return brand;
    }
       public int getwarrantyPeriod() {
```

E-Commerce System

Customer ID:
Customer Name:
Customer Address:
Select Product:
main.ElectronicProduct@56201b73
Order ID: N/A

Output - EcommerceSystem (run)

```
run:
```



E-Commerce System

Customer ID:	23012111
Customer Name:	Basmala
Customer Address:	Miami

Select Product:

main.ElectronicProduct@56201b73

```
Order ID: 3267
Order Information:
Customer ID: 23012111
Order ID: 3267
Products:
- Smartphone ($599.9)
- T-shirt ($19.99)
- OOP ($39.99)
Total Price: $659.88
```

Order ID: 3267

```
public String getbrand() {
return brand;
}
   public int getwarrantyPeriod() {
```

Output - EcommerceSystem (run)

```
run:
```