

Programming|| project

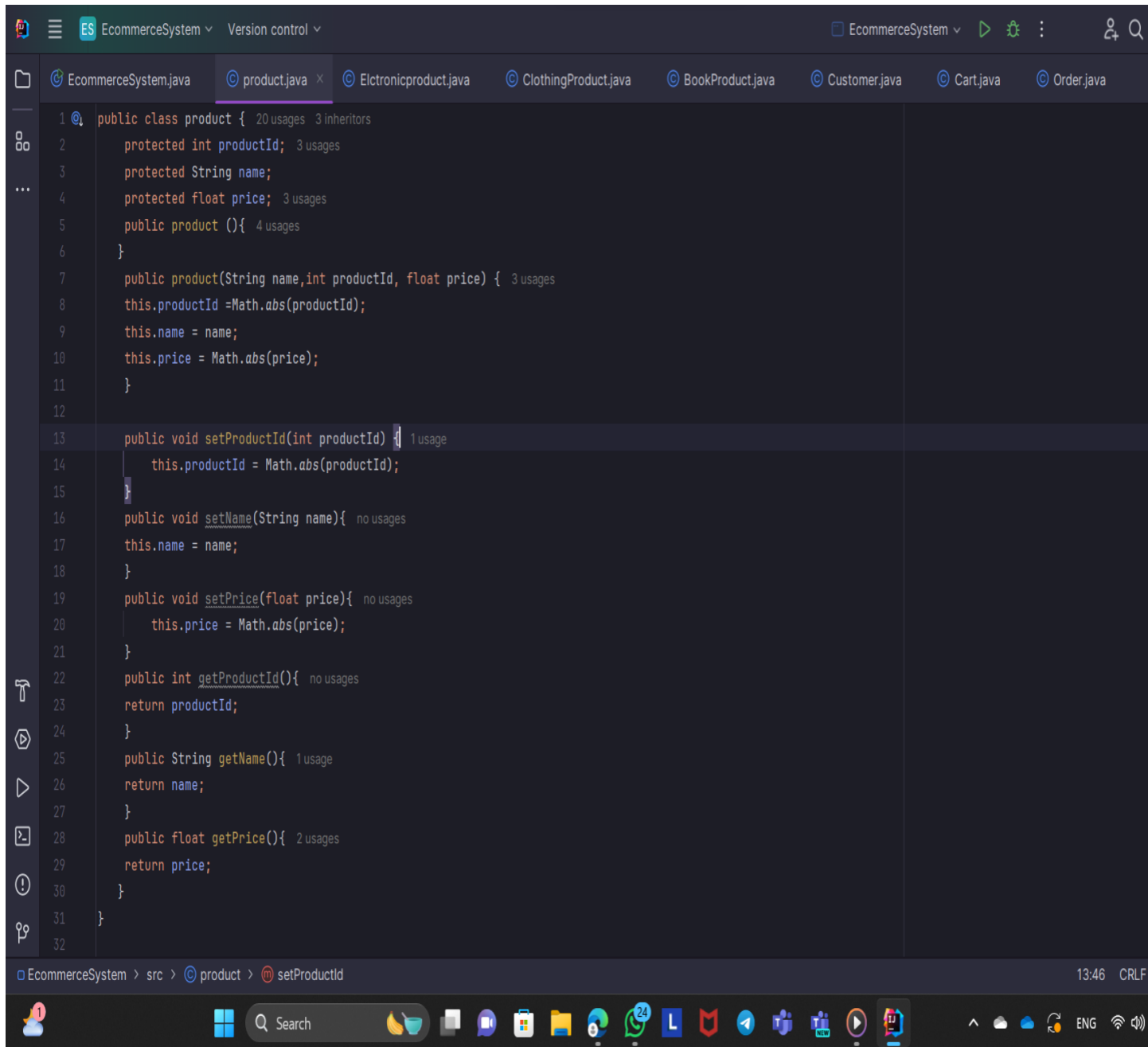
Name: Rudaina Aly Hassan

ID: 23010084

Eng: Sara Said

Classes

Product class

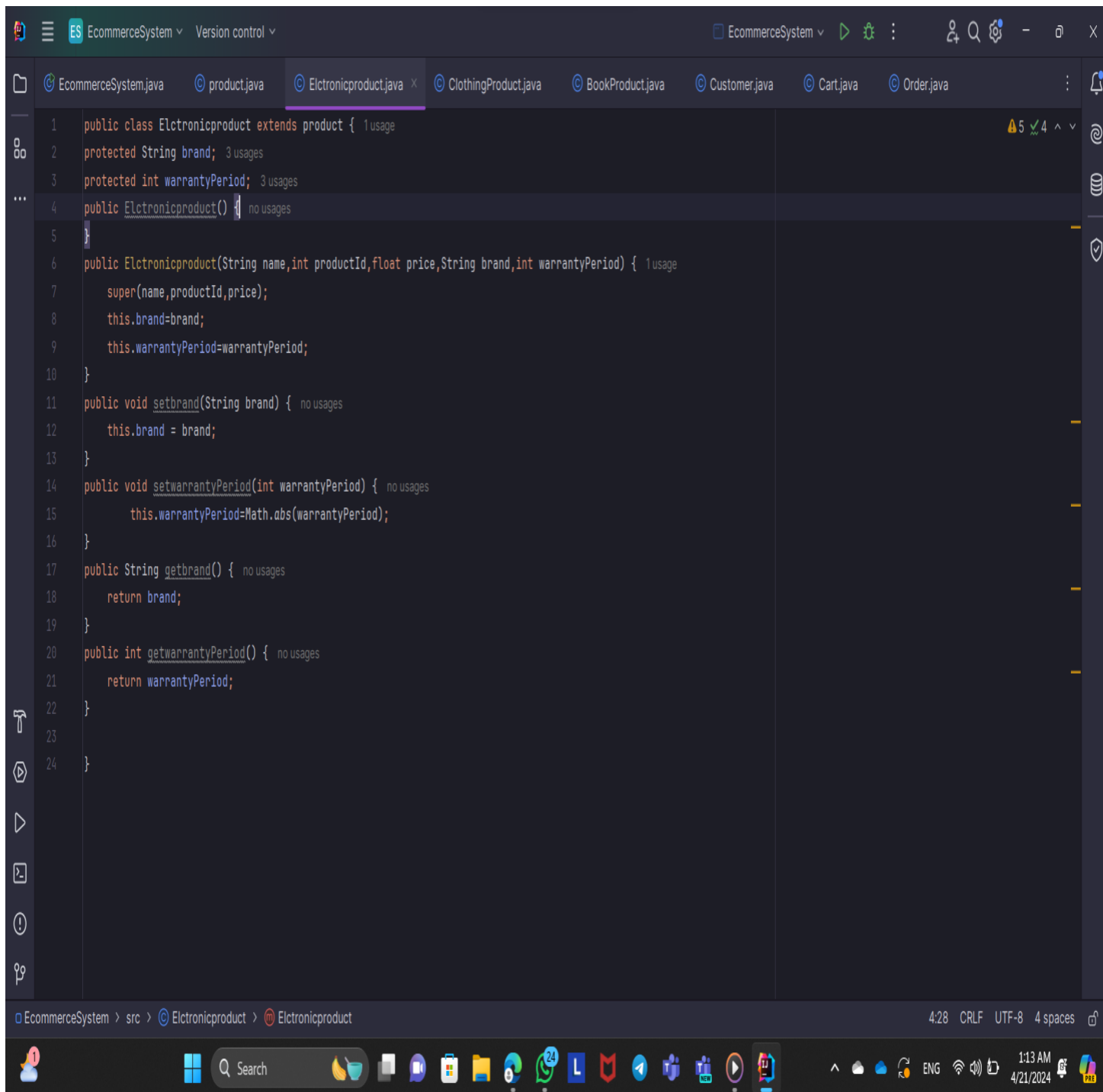


The screenshot shows an IDE window for a project named "EcommerceSystem". The "product.java" file is open, displaying the following Java code:

```
1 public class product { 20 usages 3 inheritors
2     protected int productId; 3 usages
3     protected String name;
4     protected float price; 3 usages
5     public product () { 4 usages
6     }
7     public product(String name,int productId, float price) { 3 usages
8         this.productId =Math.abs(productId);
9         this.name = name;
10        this.price = Math.abs(price);
11    }
12
13    public void setProductId(int productId) { 1 usage
14        this.productId = Math.abs(productId);
15    }
16    public void setName(String name){ no usages
17        this.name = name;
18    }
19    public void setPrice(float price){ no usages
20        this.price = Math.abs(price);
21    }
22    public int getProductId(){ no usages
23        return productId;
24    }
25    public String getName(){ 1 usage
26        return name;
27    }
28    public float getPrice(){ 2 usages
29        return price;
30    }
31 }
32
```

The IDE interface includes a top bar with the project name "EcommerceSystem" and a "Version control" dropdown. The left sidebar shows a file explorer with the project structure. The bottom status bar indicates the current file path: "EcommerceSystem > src > product > setProductId". The system tray at the bottom shows the time as 13:46 and the language as ENG.

Electronic product

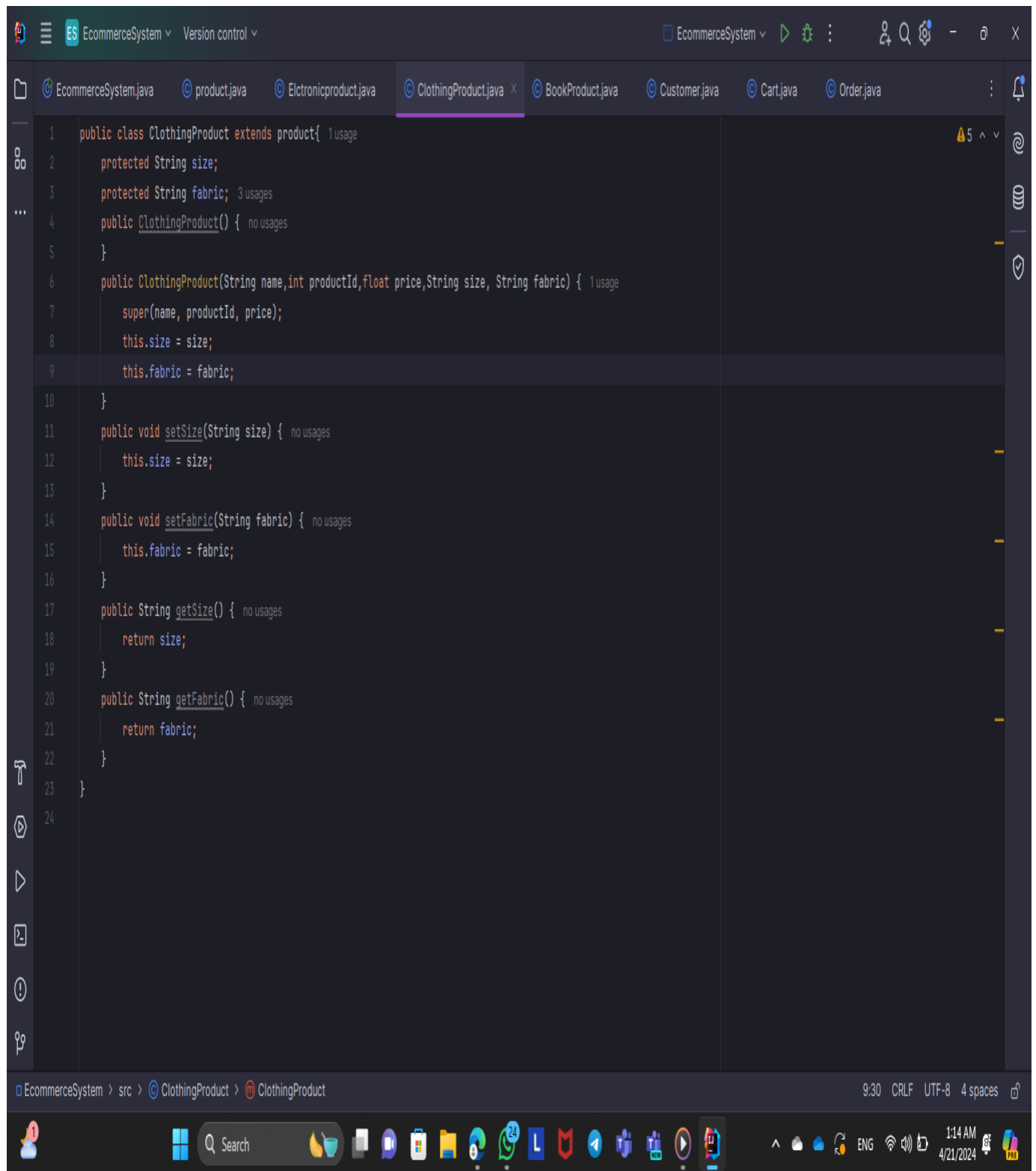


The screenshot shows an IDE window with the file `EcommerceSystem.java` open. The code defines the `Electronicproduct` class, which extends the `product` class. The class has the following attributes and methods:

- Attributes:
 - `protected String brand;` (3 usages)
 - `protected int warrantyPeriod;` (3 usages)
- Methods:
 - `public Electronicproduct()` (no usages)
 - `public Electronicproduct(String name, int productId, float price, String brand, int warrantyPeriod)` (1 usage)
 - `super(name, productId, price);`
 - `this.brand = brand;`
 - `this.warrantyPeriod = warrantyPeriod;`
 - `public void setbrand(String brand)` (no usages)
 - `this.brand = brand;`
 - `public void setwarrantyPeriod(int warrantyPeriod)` (no usages)
 - `this.warrantyPeriod = Math.abs(warrantyPeriod);`
 - `public String getbrand()` (no usages)
 - `return brand;`
 - `public int getwarrantyPeriod()` (no usages)
 - `return warrantyPeriod;`

The IDE interface includes a sidebar with icons for Explorer, Search, and Run and Debug. The bottom status bar shows the file path `EcommerceSystem > src > Electronicproduct > Electronicproduct`, the time `4:28`, the encoding `CRLF`, the character set `UTF-8`, and the indentation `4 spaces`. The Windows taskbar at the bottom shows the Start button, a search bar, and several application icons.

Clothing product

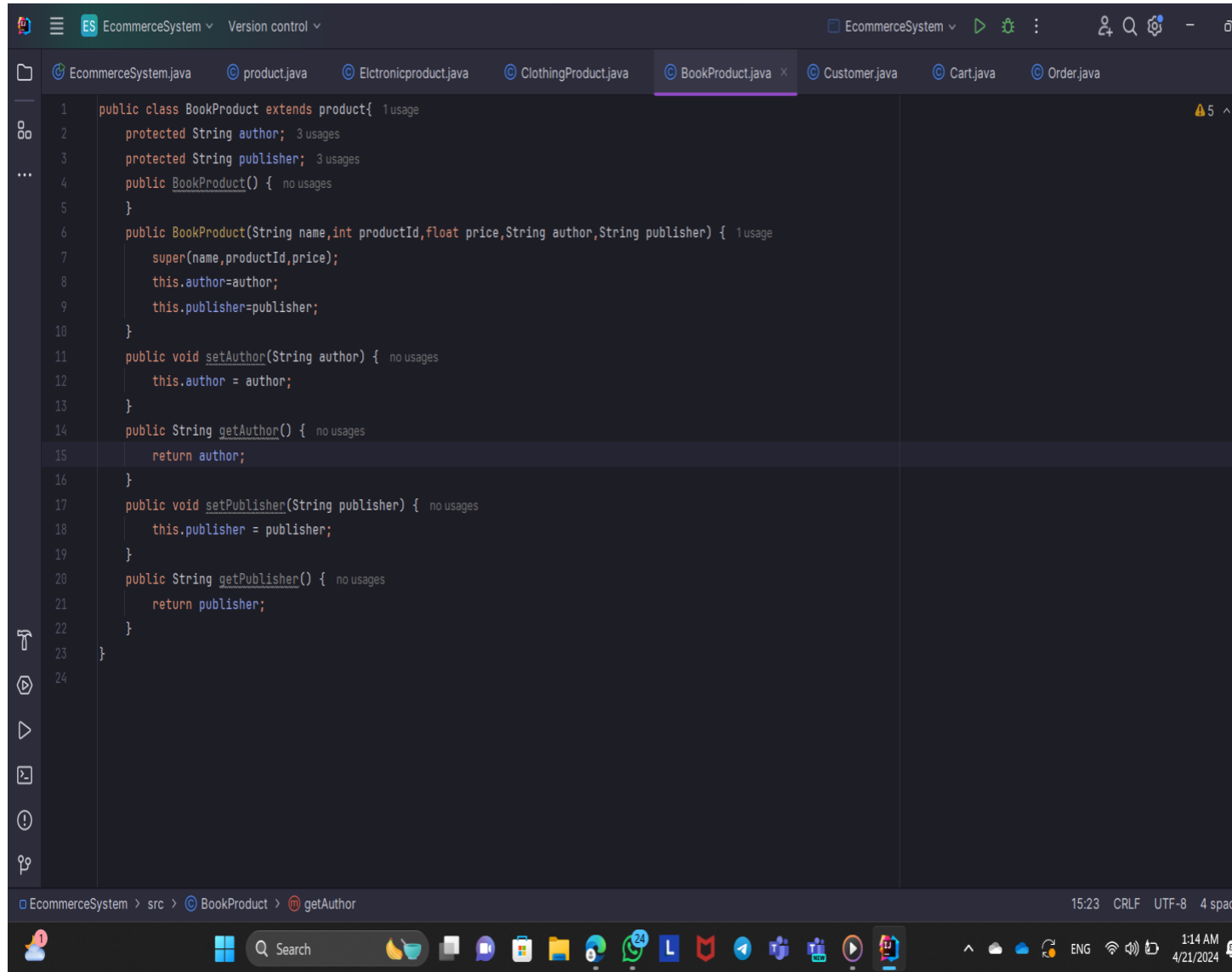


```
1 public class ClothingProduct extends Product { 1 usage
2     protected String size;
3     protected String fabric; 3 usages
4     public ClothingProduct() { no usages
5     }
6     public ClothingProduct(String name, int productId, float price, String size, String fabric) { 1 usage
7         super(name, productId, price);
8         this.size = size;
9         this.fabric = fabric;
10    }
11    public void setSize(String size) { no usages
12        this.size = size;
13    }
14    public void setFabric(String fabric) { no usages
15        this.fabric = fabric;
16    }
17    public String getSize() { no usages
18        return size;
19    }
20    public String getFabric() { no usages
21        return fabric;
22    }
23 }
24
```

EcommerceSystem > src > ClothingProduct > ClothingProduct

9:30 CRLF UTF-8 4 spaces

Book product

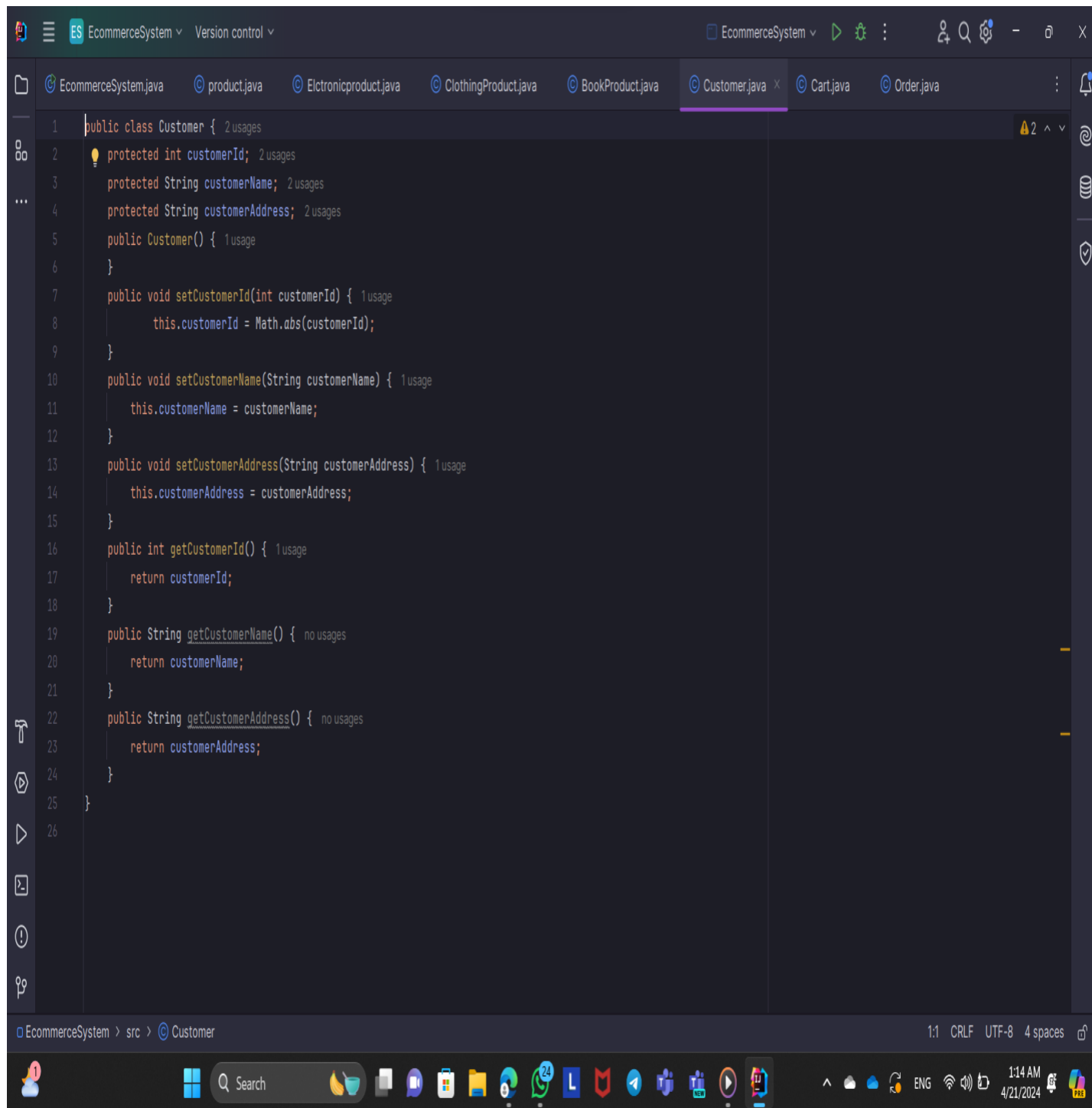


The screenshot shows a code editor with the following Java code for BookProduct.java:

```
1 public class BookProduct extends product{ 1 usage
2     protected String author; 3 usages
3     protected String publisher; 3 usages
4     public BookProduct() { no usages
5     }
6     public BookProduct(String name,int productId,float price,String author,String publisher) { 1 usage
7         super(name,productId,price);
8         this.author=author;
9         this.publisher=publisher;
10    }
11    public void setAuthor(String author) { no usages
12        this.author = author;
13    }
14    public String getAuthor() { no usages
15        return author;
16    }
17    public void setPublisher(String publisher) { no usages
18        this.publisher = publisher;
19    }
20    public String getPublisher() { no usages
21        return publisher;
22    }
23 }
24 }
```

The IDE interface includes a top toolbar with icons for file operations, a menu bar with 'ES EcommerceSystem' and 'Version control', and a tab bar showing several open files. The bottom status bar displays the file path 'EcommerceSystem > src > BookProduct > getAuthor', the time '15:23', and encoding 'CRLF UTF-8 4 spa'.

Customer

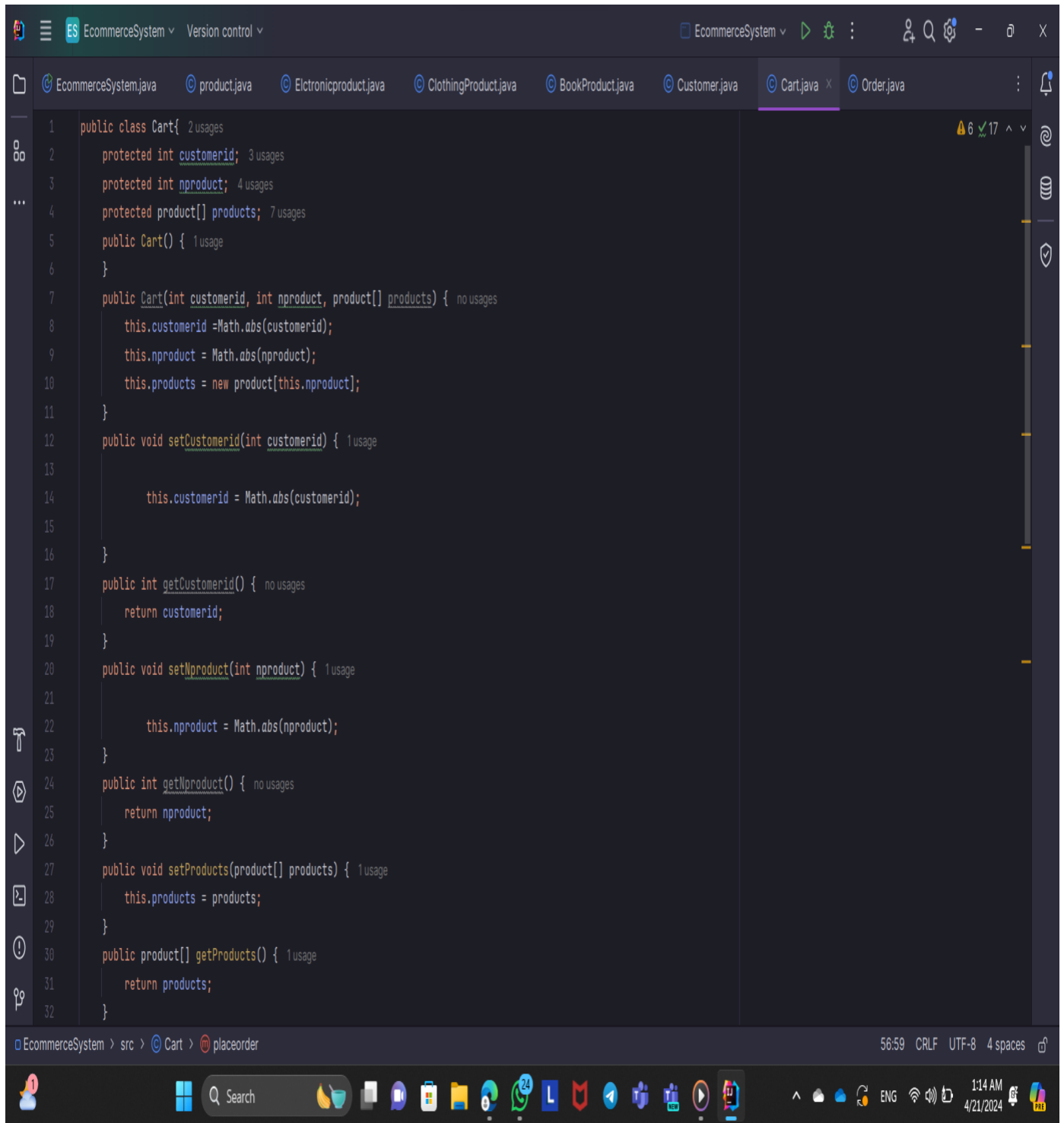


The screenshot shows a code editor with the following Java code for the Customer class:

```
1 public class Customer { 2 usages
2     protected int customerId; 2 usages
3     protected String customerName; 2 usages
4     protected String customerAddress; 2 usages
5     public Customer() { 1 usage
6     }
7     public void setCustomerId(int customerId) { 1 usage
8         this.customerId = Math.abs(customerId);
9     }
10    public void setCustomerName(String customerName) { 1 usage
11        this.customerName = customerName;
12    }
13    public void setCustomerAddress(String customerAddress) { 1 usage
14        this.customerAddress = customerAddress;
15    }
16    public int getCustomerId() { 1 usage
17        return customerId;
18    }
19    public String getCustomerName() { no usages
20        return customerName;
21    }
22    public String getCustomerAddress() { no usages
23        return customerAddress;
24    }
25 }
26
```

The IDE interface includes a top toolbar with icons for file operations, a search bar, and a status bar at the bottom showing the file path "EcommerceSystem > src > Customer", encoding "UTF-8", and line length "1:1 CRLF". The Windows taskbar at the very bottom shows the time as 1:14 AM on 4/21/2024.

Cart



The screenshot shows an IDE window with the following components:

- Top Bar:** Displays the project name "EcommerceSystem" and a "Version control" dropdown. On the right, there are icons for running, debugging, and other IDE functions.
- Tab Bar:** Shows several open files: "EcommerceSystem.java", "product.java", "Electronicproduct.java", "ClothingProduct.java", "BookProduct.java", "Customer.java", "Cart.java" (the active file), and "Order.java".
- Editor:** Contains the source code for the `Cart` class. The code is as follows:

```
1 public class Cart{ 2 usages
2     protected int customerId; 3 usages
3     protected int nproduct; 4 usages
4     protected product[] products; 7 usages
5     public Cart() { 1 usage
6     }
7     public Cart(int customerId, int nproduct, product[] products) { no usages
8         this.customerid =Math.abs(customerid);
9         this.nproduct = Math.abs(nproduct);
10        this.products = new product[this.nproduct];
11    }
12    public void setCustomerId(int customerId) { 1 usage
13
14        this.customerid = Math.abs(customerid);
15
16    }
17    public int getCustomerId() { no usages
18        returnn customerId;
19    }
20    public void setNproduct(int nproduct) { 1 usage
21
22        this.nproduct = Math.abs(nproduct);
23    }
24    public int getNproduct() { no usages
25        returnn nproduct;
26    }
27    public void setProducts(product[] products) { 1 usage
28        this.products = products;
29    }
30    public product[] getProducts() { 1 usage
31        returnn products;
32    }
```
- Bottom Bar:** Shows the file path "EcommerceSystem > src > Cart > placeorder", the time "56:59", and encoding "CRLF UTF-8 4 spaces".
- Taskbar:** At the very bottom, the Windows taskbar is visible with various application icons and the system clock showing "1:14 AM 4/21/2024".

ES EcommerceSystem Version control

Current File

⏮ ⏪ ⏩ ⏭

👤

EcommerceSystem.java product.java Electronicproduct.java ClothingProduct.java BookProduct.java Customer.java Cart.java Order.java

1 public class Cart{ 2 usages

32 }

33 public void addproduct(product product){ no usages

34 for(int i=0;i<this.products.length;i++){

35 if(this.products[i]==null){

36 this.products[i]=product;

37 break;

38 }

39 }

40 }

41 public void removeproduct(product product) { no usages

42 for(int i=0;i<this.products.length;i++){

43 if(this.products[i]==null &&this.products[i].equals(product)){

44 this.products[i]=null;

45 break;

46 }

47 }

48 }

49 public double calculateprice(){ 2 usages

50 double totalprice=0;

51 for(product product:products){

52 if (product!=null){

53 totalprice+=product.getPrice();

54 }

55 }

56 return totalprice;

57 }

58 public void placeorder(int replace){ 1 usage

59 if (replace==1){

60 System.out.println("Here's your orders's summary");

61 } else if (replace==2) {

62 System.out.println("Placement cancelled");

63 }else{

64 System.out.println("Sorry, that is not a valid ,please answer with yes or no ");

65 }

66 }

67 }

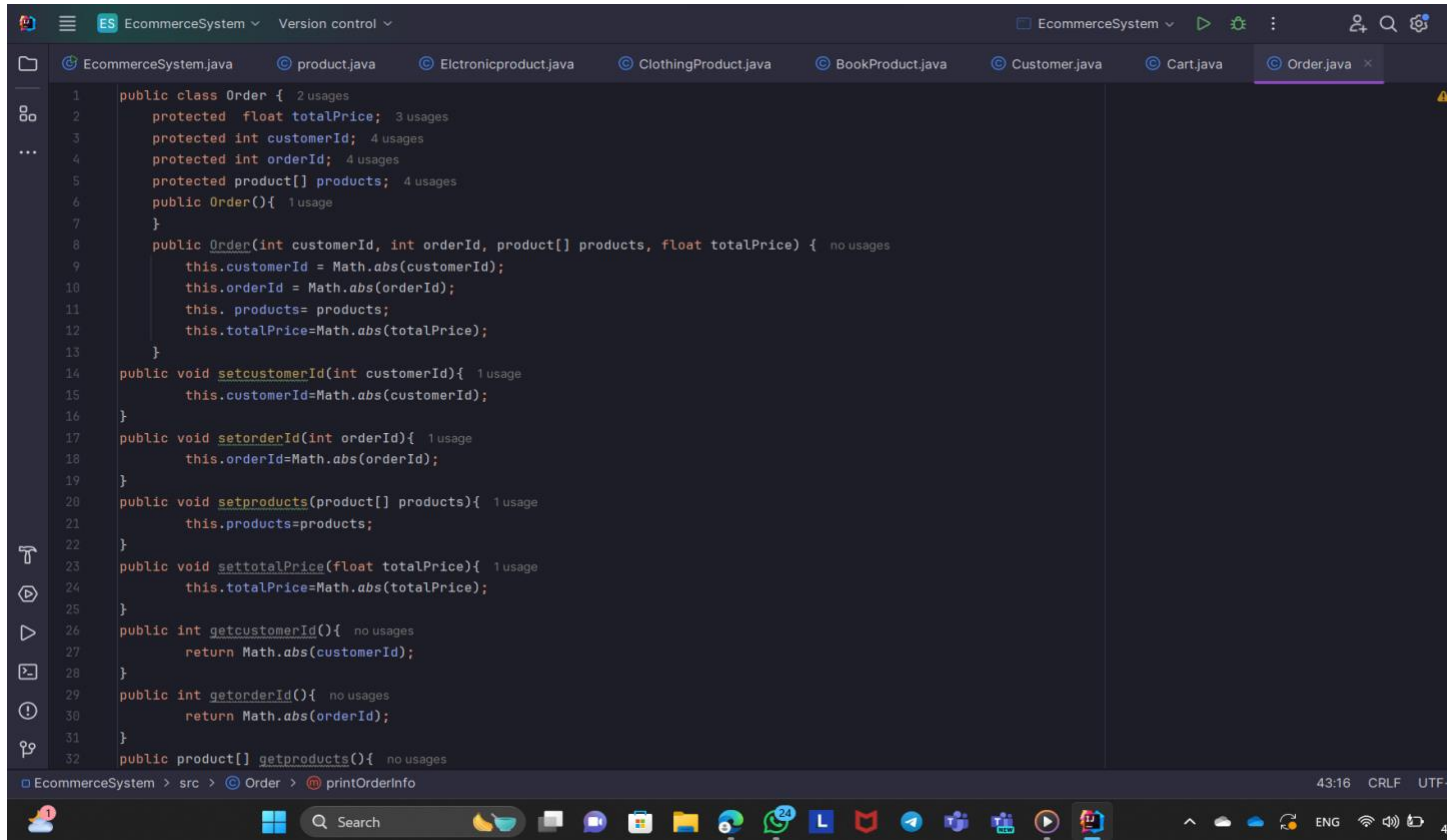
68 }

69 }

EcommerceSystem > src > Cart > calculateprice

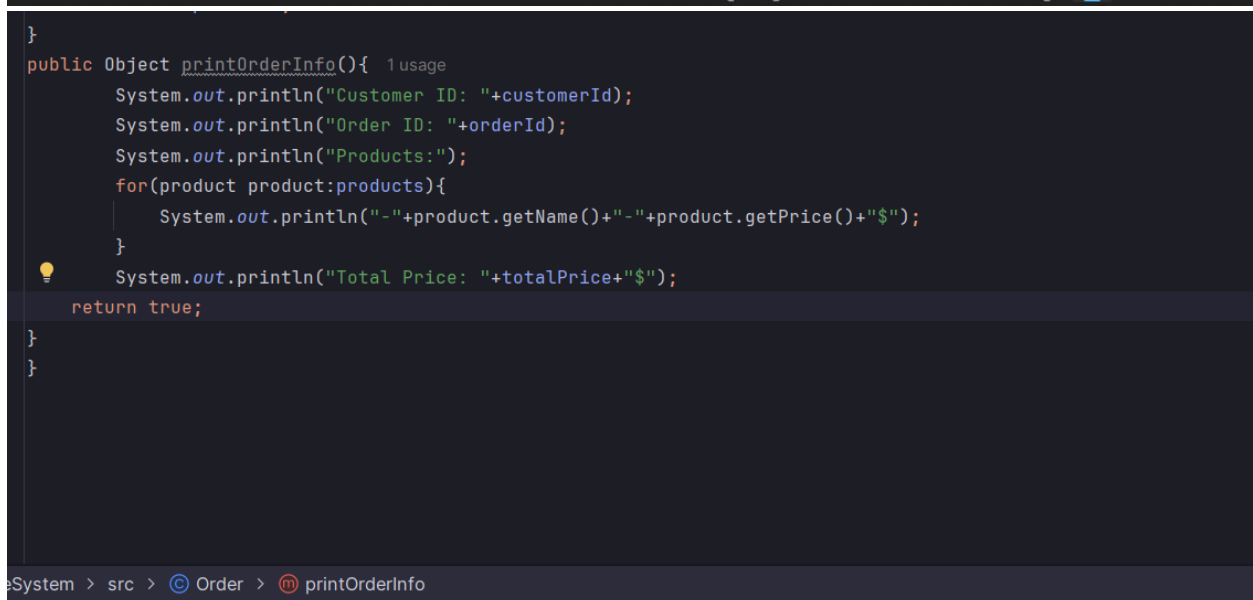
55:10 CRLF UT

Order



The screenshot shows an IDE window for a project named "EcommerceSystem". The "Order.java" file is open, displaying the following code:

```
1 public class Order { 2 usages
2     protected float totalPrice; 3 usages
3     protected int customerId; 4 usages
4     protected int orderId; 4 usages
5     protected product[] products; 4 usages
6     public Order(){ 1 usage
7     }
8     public Order(int customerId, int orderId, product[] products, float totalPrice) { no usages
9         this.customerId = Math.abs(customerId);
10        this.orderId = Math.abs(orderId);
11        this.products= products;
12        this.totalPrice=Math.abs(totalPrice);
13    }
14    public void setcustomerId(int customerId){ 1 usage
15        this.customerId=Math.abs(customerId);
16    }
17    public void setorderId(int orderId){ 1 usage
18        this.orderId=Math.abs(orderId);
19    }
20    public void setproducts(product[] products){ 1 usage
21        this.products=products;
22    }
23    public void settotalPrice(float totalPrice){ 1 usage
24        this.totalPrice=Math.abs(totalPrice);
25    }
26    public int getcustomerId(){ no usages
27        return Math.abs(customerId);
28    }
29    public int getorderId(){ no usages
30        return Math.abs(orderId);
31    }
32    public product[] getproducts(){ no usages
```

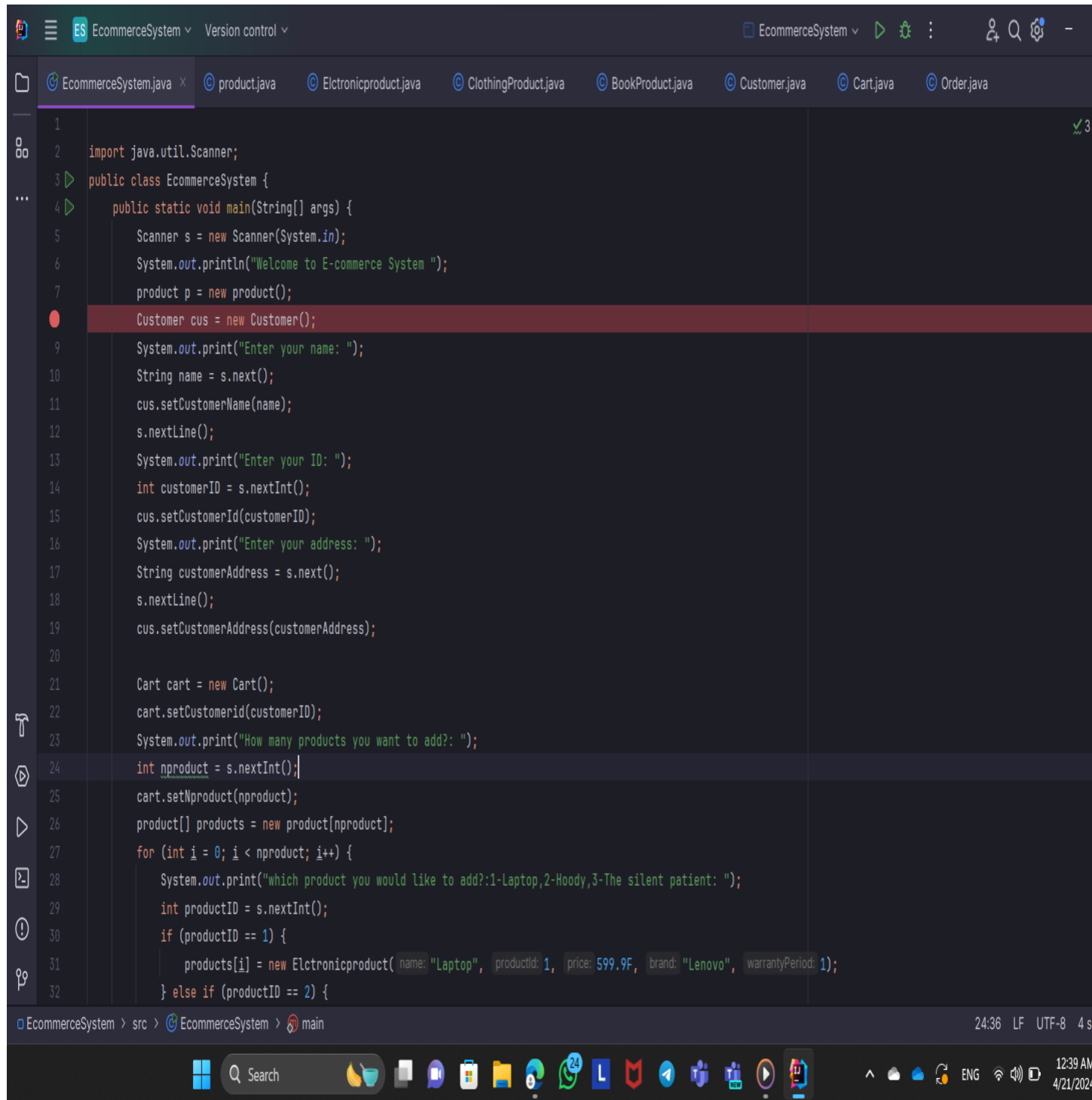


The screenshot continues from the previous one, showing the implementation of the `printOrderInfo()` method in the `Order` class:

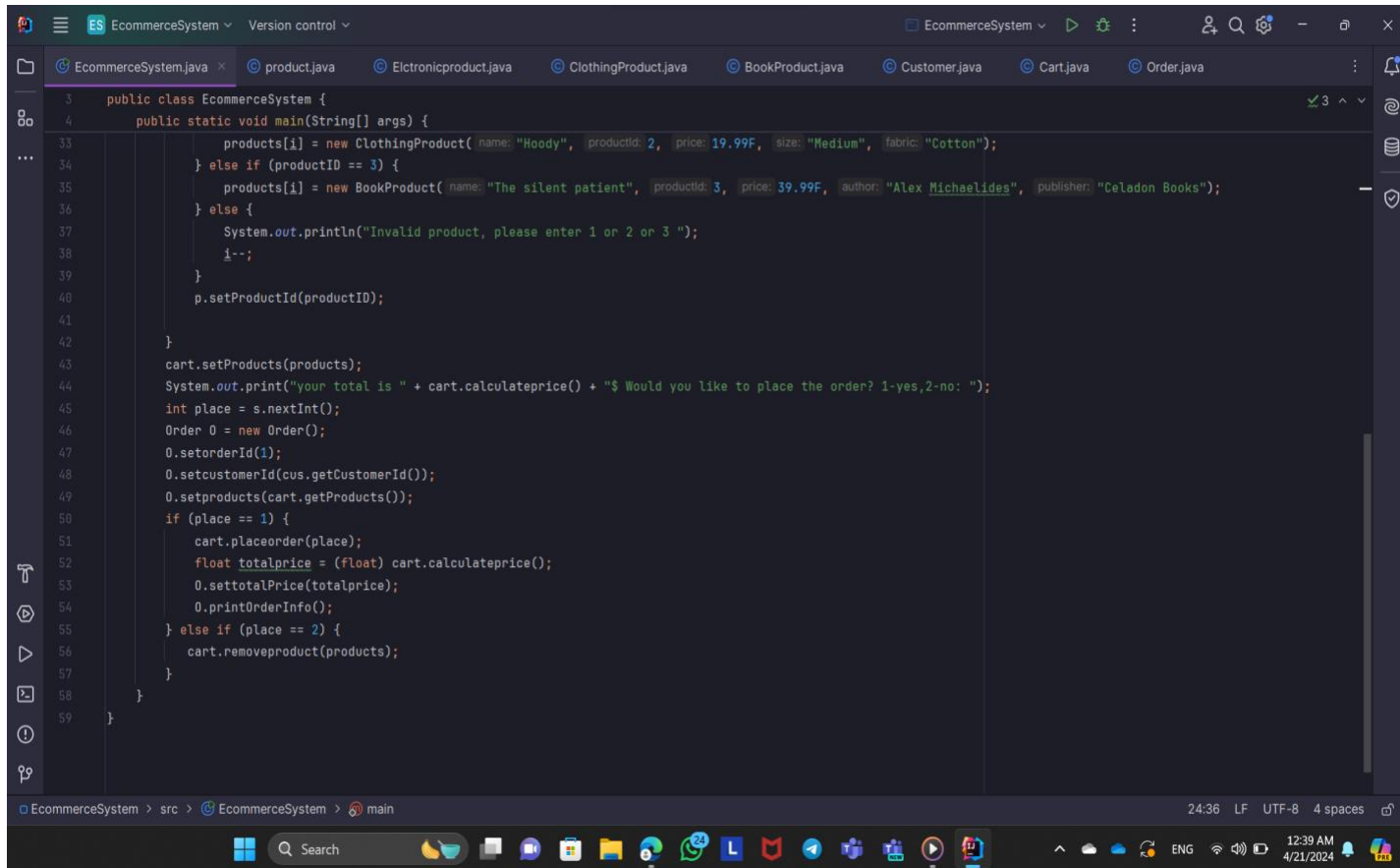
```
}
public Object printOrderInfo(){ 1 usage
    System.out.println("Customer ID: "+customerId);
    System.out.println("Order ID: "+orderId);
    System.out.println("Products:");
    for(product product:products){
        System.out.println("- "+product.getName()+"-"+product.getPrice()+"$");
    }
    System.out.println("Total Price: "+totalPrice+"$");
    return true;
}
}
```

The bottom of the IDE window shows the file explorer path: `EcommerceSystem > src > Order > printOrderInfo`.

E-commerce system(main)



```
1
2 import java.util.Scanner;
3 public class EcommerceSystem {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         System.out.println("Welcome to E-commerce System ");
7         product p = new product();
8         Customer cus = new Customer();
9         System.out.print("Enter your name: ");
10        String name = s.next();
11        cus.setCustomerName(name);
12        s.nextLine();
13        System.out.print("Enter your ID: ");
14        int customerID = s.nextInt();
15        cus.setCustomerId(customerID);
16        System.out.print("Enter your address: ");
17        String customerAddress = s.next();
18        s.nextLine();
19        cus.setCustomerAddress(customerAddress);
20
21        Cart cart = new Cart();
22        cart.setCustomerId(customerID);
23        System.out.print("How many products you want to add?: ");
24        int nproduct = s.nextInt();
25        cart.setNproduct(nproduct);
26        product[] products = new product[nproduct];
27        for (int i = 0; i < nproduct; i++) {
28            System.out.print("which product you would like to add?:1-Laptop,2-Hoody,3-The silent patient: ");
29            int productID = s.nextInt();
30            if (productID == 1) {
31                products[i] = new Elctronicproduct( name: "Laptop", productid: 1, price: 599.9F, brand: "Lenovo", warrantyPeriod: 1);
32            } else if (productID == 2) {
```



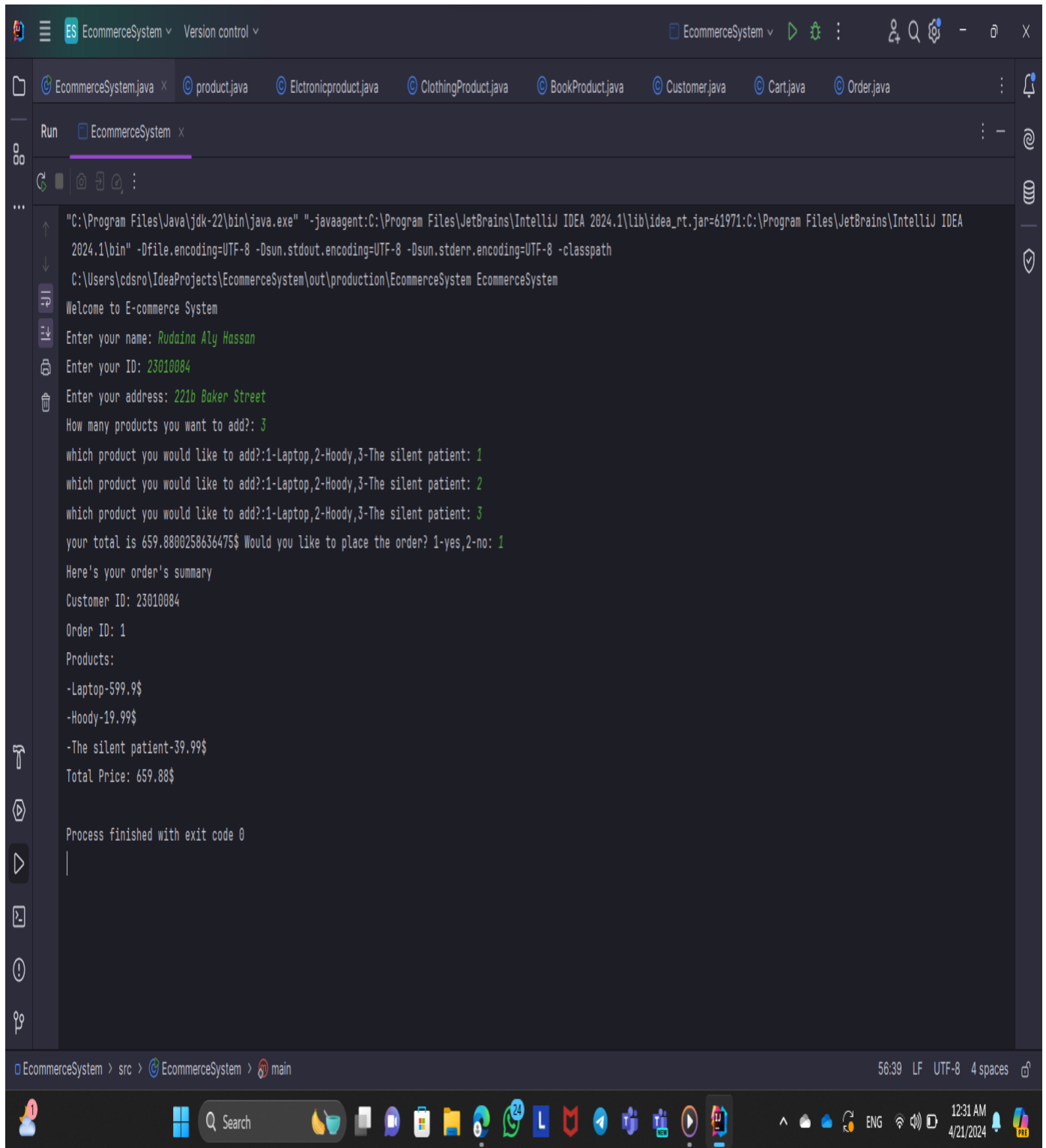
```
3 public class EcommerceSystem {
4     public static void main(String[] args) {
33         products[i] = new ClothingProduct( name: "Hoody", productid: 2, price: 19.99F, size: "Medium", fabric: "Cotton");
34     } else if (productID == 3) {
35         products[i] = new BookProduct( name: "The silent patient", productid: 3, price: 39.99F, author: "Alex Michaelides", publisher: "Celadon Books");
36     } else {
37         System.out.println("Invalid product, please enter 1 or 2 or 3 ");
38         i--;
39     }
40     p.setProductId(productID);
41
42 }
43 cart.setProducts(products);
44 System.out.print("your total is " + cart.calculateprice() + "$ Would you like to place the order? 1-yes,2-no: ");
45 int place = s.nextInt();
46 Order o = new Order();
47 o.setorderId(1);
48 o.setcustomerId(cus.getCustomerId());
49 o.setproducts(cart.getProducts());
50 if (place == 1) {
51     cart.placeorder(place);
52     float totalprice = (float) cart.calculateprice();
53     o.settotalPrice(totalprice);
54     o.printOrderInfo();
55 } else if (place == 2) {
56     cart.removeproduct(products);
57 }
58 }
59 }
```

EcommerceSystem > src > EcommerceSystem > main

24:36 LF UTF-8 4 spaces

12:39 AM 4/21/2024

Output



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\lib\idea_rt.jar=61971:C:\Program Files\JetBrains\IntelliJ IDEA 2024.1\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\cdsro\IdeaProjects\EcommerceSystem\out\production\EcommerceSystem EcommerceSystem

Welcome to E-commerce System
Enter your name: Rudaina Aly Hassan
Enter your ID: 23010084
Enter your address: 221b Baker Street
How many products you want to add?: 3
which product you would like to add?:1-Laptop,2-Hoody,3-The silent patient: 1
which product you would like to add?:1-Laptop,2-Hoody,3-The silent patient: 2
which product you would like to add?:1-Laptop,2-Hoody,3-The silent patient: 3
your total is 659.8800258636475$ Would you like to place the order? 1=yes,2=no: 1

Here's your order's summary
Customer ID: 23010084
Order ID: 1
Products:
-Laptop-599.9$
-Hoody-19.99$
-The silent patient-39.99$
Total Price: 659.88$

Process finished with exit code 0
|
```

EcommerceSystem > src > EcommerceSystem > main 56:39 LF UTF-8 4 spaces

Test case

■ *Electronic product*

Name: laptop

Price: 599.9\$

Product ID; 1

Brand: Lenovo

Warranty period: 1

■ *Clothing product*

Name: Hoody

Price: 19.9\$

Product ID:2

Size: medium

Fabric: cotton

■ *Book product*

Name: The silent patient

Price: 39.9\$

Product Id: 3

Author: Alex Michaelides

Publisher: Celadon Books

Githup link

[RudainaAly54/E_commerce-System \(github.com\)](https://github.com/RudainaAly54/E_commerce-System)