

```
EcommerceSystem master
EcommerceSystem.java Product.java ElectronicProduct.java ClothingProduct.java BookProduct.java Customer.java Cart.java Order.java

1 import java.util.Scanner;
2
3 public class EcommerceSystem { new
4     public static void main(String[] args) { new
5         System.out.println("Welcome to the Ecommerce System");
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.println("Please Enter your ID:");
9         int customerId = scanner.nextInt();
10
11         System.out.println("Enter your name:");
12         String name = scanner.next();
13
14         System.out.println("Enter your address:");
15         String address = scanner.next();
16
17         Customer c = new Customer(customerId, name, address);
18
19         // Create some example products
20         System.out.println("How many products you want to add");
21         int nProducts = scanner.nextInt();
22
23         ElectronicProduct e1 = new ElectronicProduct("Macbook", "Apple", 222, "Laptop", 200000);
24
25         ClothingProduct c1 = new ClothingProduct("Medium", "Cotton", 23, "T-shirt", 500);
26
27         BookProduct b1 = new BookProduct("O'Reilly", "X Publications", 3, "GDP", 35);
28
29         Product product;
30         int choice = scanner.nextInt();
31     }
```

```
EcommerceSystem master
EcommerceSystem.java Product.java ElectronicProduct.java ClothingProduct.java BookProduct.java Customer.java Cart.java Order.java

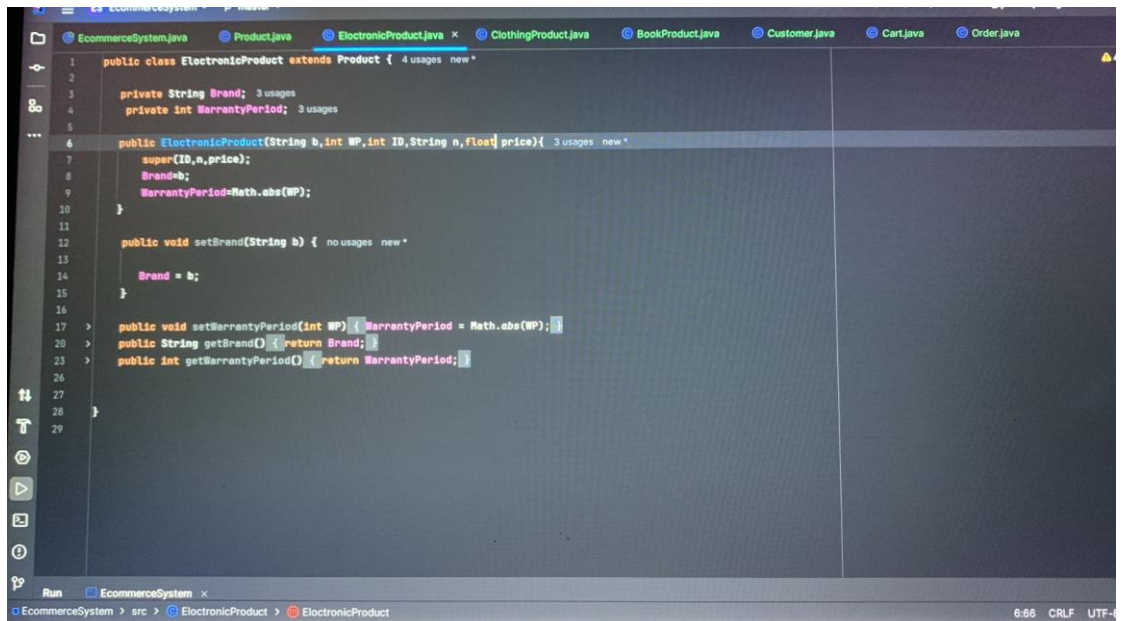
1 public class EcommerceSystem { new
2     public static void main(String[] args) { new
3
4         Product product;
5         int choice = scanner.nextInt();
6         switch (choice) {
7             case 1:
8                 product = new ElectronicProduct("Macbook", "Apple", 222, "Laptop", 200000);
9                 break;
10             case 2:
11                 product = new ClothingProduct("Medium", "Cotton", 23, "T-shirt", 500);
12                 break;
13             case 3:
14                 product = new BookProduct("O'Reilly", "X Publications", 3, "GDP", 35);
15                 break;
16             default:
17                 System.out.println("Invalid choice.");
18                 break;
19         }
20         Product[] orderedProducts = new Product[nProducts];
21
22         for (int i = 0; i < nProducts; i++) {
23             System.out.println("Enter product type (1: Electronic, 2: Clothing, 3: Book):");
24             int productType = scanner.nextInt();
25         }
26
27         Cart.addProduct(choice);
28
29         Cart c2 = new Cart();
30         System.out.println("Your total is $" + c2.calculatePrice());
31         System.out.println("Would you like to place the order? 1-yes 2-no");
32     }
```

```
46 Product[] orderedProducts = new Product[nProducts];
47
48 for (int i = 0; i < nProducts; i++) {
49     System.out.println("Enter product type (1: Electronic, 2: Clothing, 3: Book):");
50     int productType = scanner.nextInt();
51 }
52
53 Cart.addProduct(choice);
54
55
56 Cart c2 = new Cart();
57 System.out.println("Your total is $" + c2.calculatePrice());
58 System.out.println("Would you like to place the order? 1=yes 2=no");
59 int placeOrderChoice = scanner.nextInt();
60 if (placeOrderChoice == 1) {
61     System.out.println("Here's your order summary:");
62     Order o1 = new Order(c2.getOrderID());
63     o1.printOrderInfo();
64 } else {
65     System.out.println("Order not placed.");
66 }
67 }
68 }
```

Run EcommerceSystem x
EcommerceSystem > src > EcommerceSystem > main

```
1 public class Product { 15 usages 3 inherits new *
2
3     private int ProductID; 3 usages
4     private String name; 3 usages
5     private float Price; 5 usages
6
7
8
9
10    public Product(int ID,String n,float price) 3 usages new *
11    {
12        ProductID=Math.abs(ID);
13        name=n;
14        Price=Math.abs(Price);
15    }
16
17    public void setProductID(int ID){ ProductID =Math.abs(ID); }
18
19    public void setName(String n){ name = n; }
20
21    public void setPrice(float price) { no usages new *
22        Price = Math.abs(Price);
23    }
24
25    public int getProductID(){ return ProductID; }
26
27    public String getName(){ return name; }
28
29    public float getPrice(){ return Price; }
30
31
32
33
34
35
36
37
38
39
40 }
```

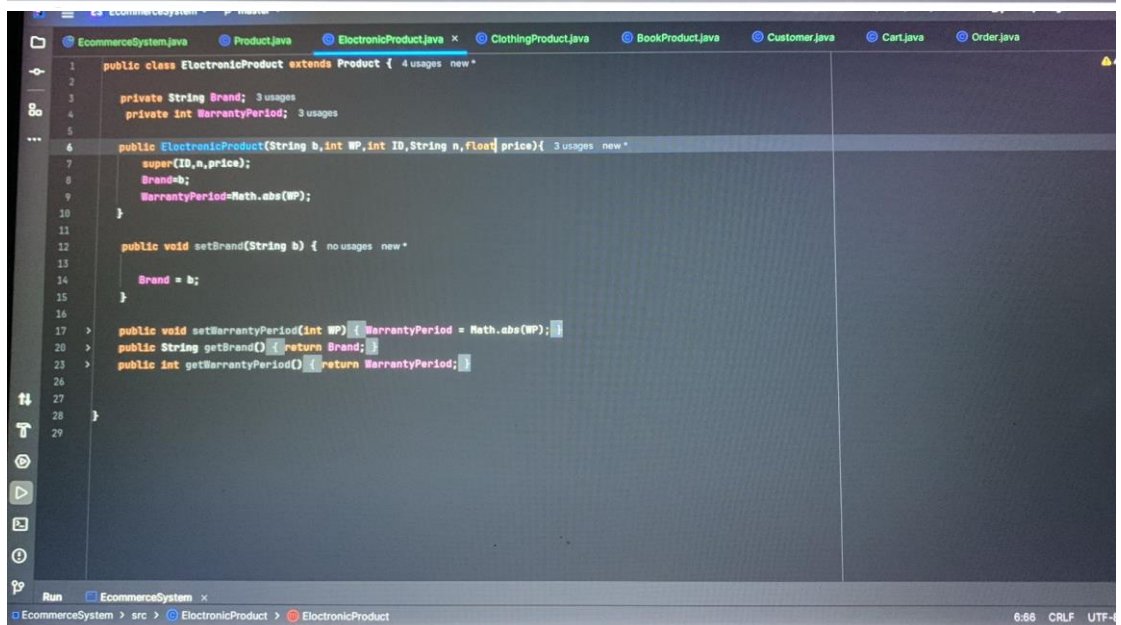
Run EcommerceSystem x
EcommerceSystem > src > Product 38:5 CRLF UTF-8 4 spaces



```
1 public class ElectronicProduct extends Product { 4 usages new *
2
3     private String Brand; 3 usages
4     private int WarrantyPeriod; 3 usages
5
6     public ElectronicProduct(String b,int WP,int ID,String n,float price){ 3 usages new *
7         super(ID,n,price);
8         Brand=b;
9         WarrantyPeriod=Math.abs(WP);
10    }
11
12    public void setBrand(String b) { no usages new *
13
14        Brand = b;
15    }
16
17    public void setWarrantyPeriod(int WP){ WarrantyPeriod = Math.abs(WP); }
18
19    public String getBrand() { return Brand; }
20
21    public int getWarrantyPeriod() { return WarrantyPeriod; }
22
23
24
25
26
27
28
29 }
```

Run ☒ EclipseSystem x

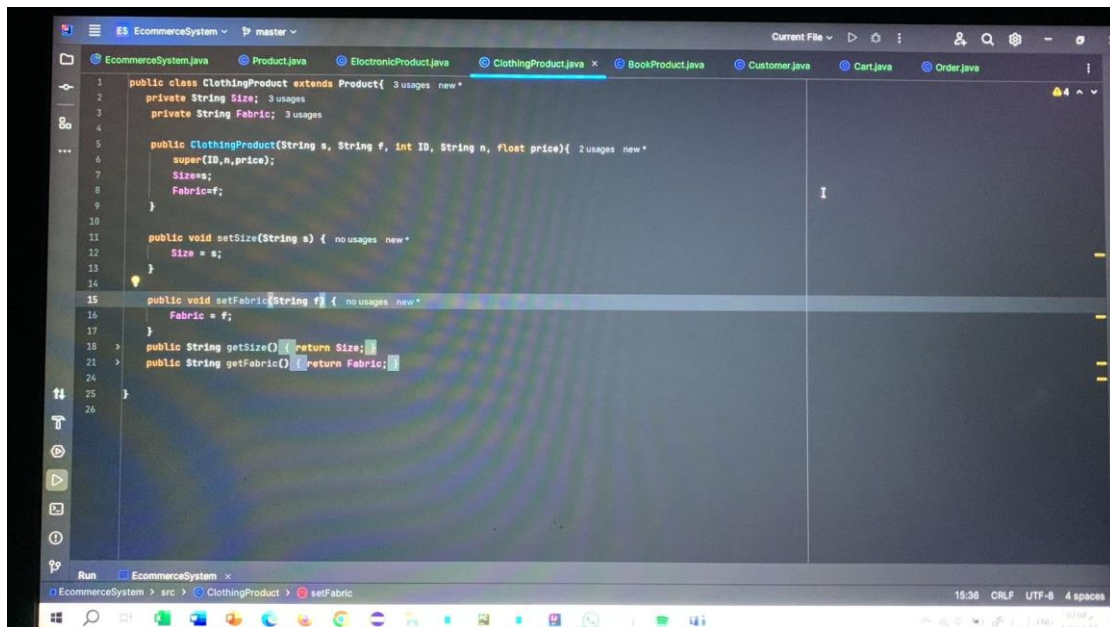
EcommerceSystem > src > ElectronicProduct > ElectronicProduct 6:55 CRLF UTF-8



```
1 public class ElectronicProduct extends Product { 4 usages new *
2
3     private String Brand; 3 usages
4     private int WarrantyPeriod; 3 usages
5
6     public ElectronicProduct(String b,int WP,int ID,String n,float price){ 3 usages new *
7         super(ID,n,price);
8         Brand=b;
9         WarrantyPeriod=Math.abs(WP);
10    }
11
12    public void setBrand(String b) { no usages new *
13
14        Brand = b;
15    }
16
17    public void setWarrantyPeriod(int WP){ WarrantyPeriod = Math.abs(WP); }
18
19    public String getBrand() { return Brand; }
20
21    public int getWarrantyPeriod() { return WarrantyPeriod; }
22
23
24
25
26
27
28
29 }
```

Run ☒ EclipseSystem x

EcommerceSystem > src > ElectronicProduct > ElectronicProduct 6:55 CRLF UTF-8

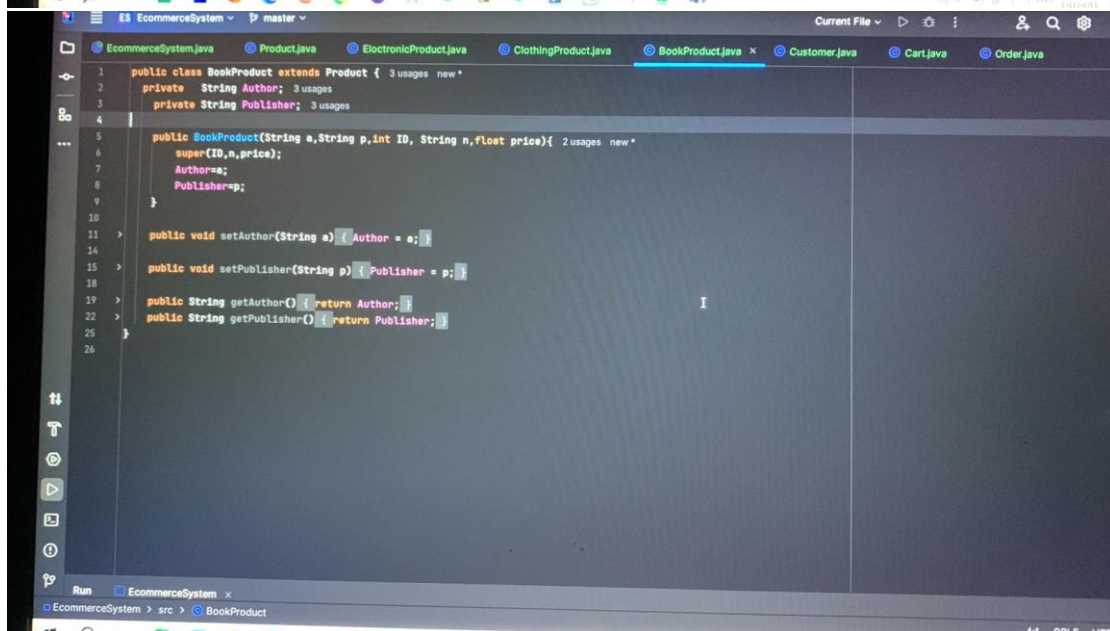


```
1 public class ClothingProduct extends Product { 3 usages new *
2     private String Size; 3 usages
3     private String Fabric; 3 usages
4
5     public ClothingProduct(String s, String f, int ID, String n, float price) { 2 usages new *
6         super(ID, n, price);
7         Size = s;
8         Fabric = f;
9     }
10
11     public void setSize(String s) { no usages new *
12         Size = s;
13     }
14
15     public void setFabric(String f) { no usages new *
16         Fabric = f;
17     }
18     public String getSize() { return Size; }
19     public String getFabric() { return Fabric; }
20 }
21
22
23
24
25
26
```

Run EcommerceSystem x

EcommerceSystem > src > ClothingProduct > setFabric

15:38 CRLF UTF-8 4 spaces



```
1 public class BookProduct extends Product { 3 usages new *
2     private String Author; 3 usages
3     private String Publisher; 3 usages
4
5     public BookProduct(String a, String p, int ID, String n, float price) { 2 usages new *
6         super(ID, n, price);
7         Author = a;
8         Publisher = p;
9     }
10
11     public void setAuthor(String a) { Author = a; }
12
13     public void setPublisher(String p) { Publisher = p; }
14
15     public String getAuthor() { return Author; }
16     public String getPublisher() { return Publisher; }
17 }
18
19
20
21
22
23
24
25
26
```

Run EcommerceSystem x

EcommerceSystem > src > BookProduct

4:1 CRLF UTF-8

```
1 public class Cart { 3 usages new *
2     private int customerId; 3 usages
3     private int nProducts; 5 usages
4     private Product[] products; 7 usages
5
6     // Constructor
7     public Cart(int customerId, int nProducts) { no usages new *
8         this.customerId = Math.abs(customerId); // Take absolute value to ensure positive integer
9         this.nProducts = Math.abs(nProducts); // Take absolute value to ensure positive integer
10        this.products = new Product[nProducts];
11    }
12
13    public Cart() { 1 usage new *
14    }
15
16    public static void addProduct(int product) { 1 usage new *
17    }
18
19
20    public int getCustomerId() { no usages new *
21        return customerId;
22    }
23
24    public void setCustomerId(int customerId) { no usages new *
25        this.customerId = customerId;
26    }
27
28    public Product[] getProducts() { no usages new *
29        return products;
30    }
31 }
```

Run EcommerceSystem x
EcommerceSystem > src > Cart > calculatePrice

64:1 CRLF UTF-8 4 spaces

```
1 public class Cart { 3 usages new *
2     private int customerId; 3 usages
3     private int nProducts; 5 usages
4     private Product[] products; 7 usages
5
6     // Constructor
7     public Cart(int customerId, int nProducts) { no usages new *
8         this.customerId = Math.abs(customerId); // Take absolute value to ensure positive integer
9         this.nProducts = Math.abs(nProducts); // Take absolute value to ensure positive integer
10        this.products = new Product[nProducts];
11    }
12
13    public Cart() { 1 usage new *
14    }
15
16    public static void addProduct(int product) { 1 usage new *
17    }
18
19
20    public int getCustomerId() { no usages new *
21        return customerId;
22    }
23
24    public void setCustomerId(int customerId) { no usages new *
25        this.customerId = customerId;
26    }
27
28    public Product[] getProducts() { no usages new *
29        return products;
30    }
31 }
```

Run EcommerceSystem x
EcommerceSystem > src > Cart > calculatePrice

64:1 CRLF UTF-8 4 spaces

```
1 public class Cart { 3 usages new *
31
32     public void setProducts(Product[] products) { no usages new *
33         this.products = products;
34     }
35
36     public int getProducts() { return nProducts; }
37
38     public void setnProducts(int nProducts) { no usages new *
39         this.nProducts = nProducts;
40     }
41
42     // methode to add a product to the cart
43     public void addProduct(Product product, int index) { no usages new *
44         if (index >= 0 && index < nProducts) {
45             products[index] = product;
46         } else {
47             System.out.println("the product has been added to cart.");
48         }
49     }
50
51     //methode to remove a product from the cart
52     public void removeProduct(int index){ no usages new *
53         if (index >= 0 && index < nProducts) {
54             products[index] = null;
55         } else {
56             System.out.println("the product has been removed from cart.");
57         }
58     }
59
60     //methode to calc the total price of all products in the cart
61     public double calculatePrice() { 1 usage new *
62
63         double Price = 0;
64         for (Product product : products) {
65             if (product != null) {
66                 Price += product.getPrice();
67             }
68         }
69         return Price;
70     }
71
72 }
73
```



```
EcommerceSystem.java Product.java ElectronicProduct.java ClothingProduct.java BookProduct.java Customer.java Cart.java

1 public class Order { 2 usages new *
2     private int customerId; 2 usages
3     private int orderId; 3 usages
4     private Product[] products; 3 usages
5     private float totalPrice; 2 usages
6     // Constructor
7     public Order(int customerId, int orderId, Product[] products, float totalPrice) { no usages new *
8         this.customerId = Math.abs(customerId); // Take absolute value to ensure positive integer
9         this.orderId = Math.abs(orderId); // Take absolute value to ensure positive integer
10        this.products = products;
11        this.totalPrice = Math.abs(totalPrice); // Take absolute value to ensure positive float
12    }
13    public Order(int orderId) { 1 usage new *
14        this.orderId = orderId;
15    }
16    public int getOrderId() { 1 usage new *
17        return orderId;
18    }
19    //method to print order info.
20    public void printOrderInfo(){ 1 usage new *
21        System.out.println("here is your order summary");
22        System.out.println("Customer ID: " + customerId);
23        System.out.println("Order ID: " + getOrderId());
24        System.out.println("Products:");
25        for (Product product : products) {
26            if (product != null) {
27                System.out.println("    " + product.getName() + " - $" + product.getPrice());
28            }
29        }
30        System.out.println("Total Price: $" + totalPrice);
31    }
}
```

Run EcommerceSystem x 8:30 CRLF UTF-8 4 sp

EcommerceSystem > src > Order > totalPrice