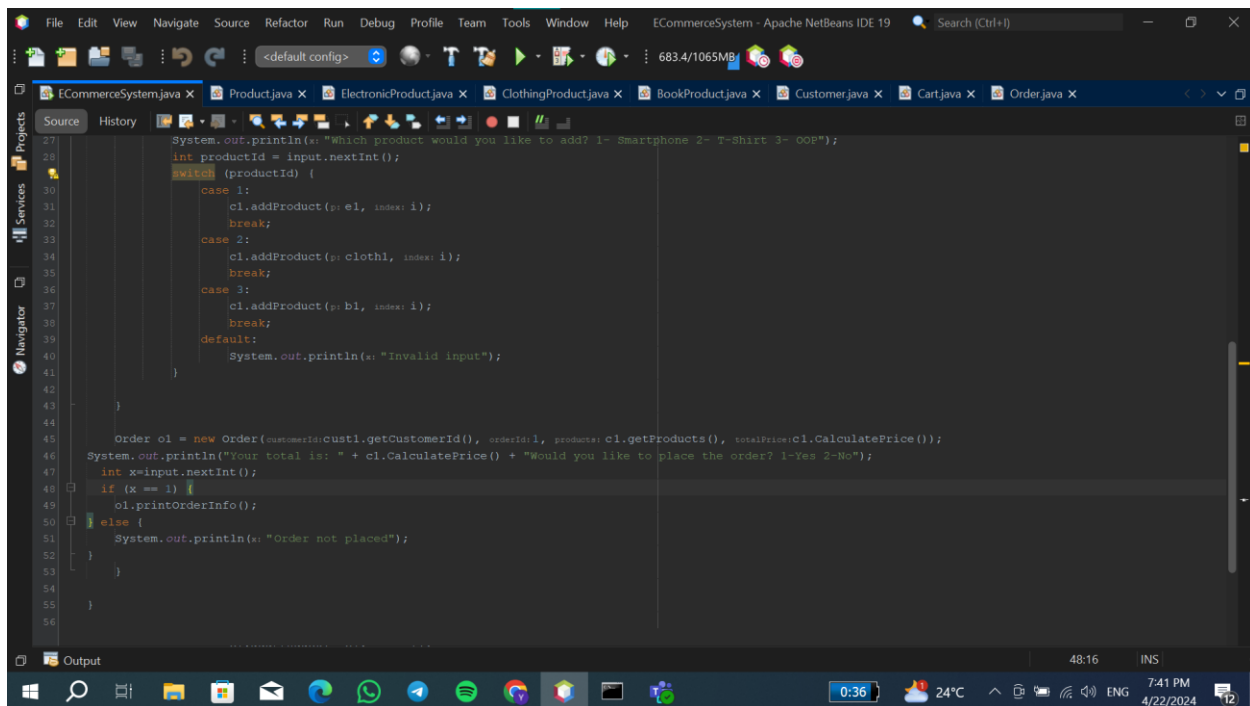


```
1 package ecommercesystem;
2 import java.util.Scanner;
3 public class ECommerceSystem {
4
5     public static void main(String[] args) {
6         Scanner input = new Scanner(System.in);
7         ElectronicProduct e1 = new ElectronicProduct("Samsung", "warrantyPeriod:1", "ProductId: 1", "name: 'Smartphone'", (float)599.9);
8         ClothingProduct cloth1 = new ClothingProduct("size: 'Medium'", "fabric: 'Cotton'", "ProductId: 2", "name: 'T-shirt'", (float)19.99);
9         BookProduct b1 = new BookProduct("author: 'O'Reilly'", "publisher: 'X Publications'", "ProductId: 3", "name: 'OOP'", (float)39.99);
10
11         System.out.println("Welcome to the E-Commerce system!");
12         System.out.println("Please enter your id ");
13         int id = input.nextInt();
14         System.out.println("Please enter your name ");
15         String n = input.next();
16         System.out.println("Please enter your address ");
17         String a = input.next();
18         Customer cust1 = new Customer(customerId: id, name: n, address: a);
19
20         System.out.println("How many products do you want to add to your cart? ");
21         int nProducts = input.nextInt();
22         Product[] products = new Product[nProducts];
23         Cart c1 = new Cart(customerId: id, nProducts, products);
24
25         for (int i = 0; i < nProducts; i++) {
26             System.out.println("Which product would you like to add? 1- Smartphone 2- T-Shirt 3- OOP");
27             int productId = input.nextInt();
28             switch (productId) {
29                 case 1:
30                     c1.addProduct(e1, index: i);
31             }
32         }
33     }
34 }
```



```
27         System.out.println("Which product would you like to add? 1- Smartphone 2- T-Shirt 3- OOP");
28         int productId = input.nextInt();
29         switch (productId) {
30             case 1:
31                 c1.addProduct(p: e1, index: i);
32                 break;
33             case 2:
34                 c1.addProduct(p: cloth1, index: i);
35                 break;
36             case 3:
37                 c1.addProduct(p: b1, index: i);
38                 break;
39             default:
40                 System.out.println("Invalid input");
41         }
42     }
43
44     Order o1 = new Order(customerId: cust1.getCustomerId(), orderId: 1, products: c1.getProducts(), totalPrice: c1.CalculatePrice());
45     System.out.println("Your total is: " + c1.CalculatePrice() + "Would you like to place the order? 1-Yes 2-No");
46     int x = input.nextInt();
47     if (x == 1) {
48         o1.printOrderInfo();
49     } else {
50         System.out.println("Order not placed");
51     }
52 }
53
54 }
55
56 }
```

This screenshot shows the NetBeans IDE with the `Product.java` file open. The code defines a `Product` class within the `ecommercesystem` package. It includes protected attributes for `ProductId`, `name`, and `price`. The class has a constructor and several methods: `setProductId`, `setName`, `setPrice`, `getProductId`, `getName`, and `getPrice`. The `set` methods use `Math.abs` to ensure non-negative values. The IDE interface includes a menu bar, a toolbar, a project explorer on the left, and a status bar at the bottom showing the time as 7:42 PM on 4/22/2024.

```
1 package ecommercesystem;
2 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 = ProductId;
9         this.name = name;
10        this.price = price;
11    }
12
13    public void setProductId(int ProductId) {
14        ProductId=Math.abs(ProductId);
15        this.ProductId = ProductId;
16    }
17
18    public void setName(String name) {
19        this.name = name;
20    }
21
22    public void setPrice(float price) {
23        price=Math.abs(price);
24        this.price = price;
25    }
26
27    public int getProductId() {
28        return ProductId;
29    }
30
31    public String getName() {
32        return name;
33    }
34
35    public float getPrice() {
36        return price;
37    }
38 }
```

This screenshot shows the NetBeans IDE with the `ElectronicProduct.java` file open. The code defines the `ElectronicProduct` class, which extends the `Product` class. It adds private attributes for `brand` and `warrantyPeriod`. The class includes a constructor and methods: `setBrand`, `setWarrantyPeriod`, `getBrand`, and `getWarrantyPeriod`. The `setWarrantyPeriod` method uses `Math.abs` to ensure a non-negative value. The IDE interface is similar to the previous screenshot, with the status bar at the bottom showing the time as 7:43 PM on 4/22/2024.

```
1 package ecommercesystem;
2 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 = warrantyPeriod;
10    }
11
12
13    public void setBrand(String brand) {
14        this.brand = brand;
15    }
16
17
18    public void setWarrantyPeriod(int warrantyPeriod) {
19        warrantyPeriod= Math.abs(warrantyPeriod);
20        this.warrantyPeriod = warrantyPeriod;
21    }
22
23    public String getBrand() {
24        return brand;
25    }
26
27    public int getWarrantyPeriod() {
28        return warrantyPeriod;
29    }
30 }
```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ECommerceSystem - Apache NetBeans IDE 19 Search (Ctrl+I)

<default config> 489.9/1065MB

ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x

Source History

```
1 package ecommercesystem;
2 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 void setSize(String size) {
13        this.size = size;
14    }
15
16    public void setFabric(String fabric) {
17        this.fabric = fabric;
18    }
19
20    public String getSize() {
21        return size;
22    }
23
24    public String getFabric() {
25        return fabric;
26    }
27 }
```

Output 12:26 INS Windows (CRLF)

0:34 24°C 7:43 PM 4/22/2024

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ECommerceSystem - Apache NetBeans IDE 19 Search (Ctrl+I)

<default config> 798.9/1065MB

ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x

Source History

```
1 package ecommercesystem;
2 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 void setAuthor(String author) {
13        this.author = author;
14    }
15
16    public void setPublisher(String publisher) {
17        this.publisher = publisher;
18    }
19
20    public String getAuthor() {
21        return author;
22    }
23
24    public String getPublisher() {
25        return publisher;
26    }
27 }
```

Output 24:1 INS Windows (CRLF)

0:33 24°C 7:44 PM 4/22/2024

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates the project is 'ECommerceSystem' and the IDE version is 'Apache NetBeans IDE 19'. The toolbar shows various icons for file operations, running, and debugging. The project explorer on the left shows the 'ECommerceSystem' project with a 'Source' folder containing several Java files. The main editor window displays the 'Customer.java' file with the following code:

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

The status bar at the bottom shows '30:2' and 'INS Windows (CRLF)'.

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates the project is 'ECommerceSystem' and the IDE version is 'Apache NetBeans IDE 19'. The toolbar shows various icons for file operations, running, and debugging. The project explorer on the left shows the 'ECommerceSystem' project with a 'Source' folder containing several Java files. The main editor window displays the 'Cart.java' file with the following code:

```
1 package ecommerceSystem;
2 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 = customerId;
9         this.nProducts = nProducts;
10        this.products = products;
11    }
12
13    public void setCustomerId(int customerId) {
14        customerId=Math.abs(+ customerId);
15        this.customerId = customerId;
16    }
17    public void setnProducts(int nProducts) {
18        nProducts=Math.abs(+ nProducts);
19        this.nProducts = nProducts;
20    }
21    public void setProducts(Product products[]){
22        this.products = new Product[nProducts];
23    }
24    public int getCustomerId() {
25        return customerId;
26    }
27    public int getnProducts() {
28        return nProducts;
29    }
30    public Product[] getProducts() {
31        return products;
32    }
33 }
```

The status bar at the bottom shows '31:25' and 'INS Windows (CRLF)'.

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ECommerceSystem - Apache NetBeans IDE 19 Search (Ctrl+I)
<default config> 722.9/1065MB
ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x
Source History
30 public Product[] getProducts() {
31     return products;
32 }
33 public void addProduct(Product p, int index) {
34     if (index >= 0 && index < nProducts)
35         products[index] = p;
36     else
37         System.out.println(s: "Invalid");
38 }
39
40 public void removeProduct(int index){
41     if (index >= 0 && index < nProducts)
42         products[index] = null;
43     else
44         System.out.println(s: "Invalid");
45 }
46
47
48 public float CalculatePrice(){
49     float price=0;
50     for (int i=0; i<nProducts; i++){
51         price += products[i].getPrice();
52     }
53     return price;
54 }
55
56 public void placeOrder(){
57     Order o = new Order(customerId, orderId, products, totalPrice:CalculatePrice());
58 }
59 }
Output 60:1 INS Windows (CRLF)
0:32 24°C 7:46 PM 4/22/2024
```

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ECommerceSystem - Apache NetBeans IDE 19 Search (Ctrl+I)
<default config> 586.9/1065MB
ECommerceSystem.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x
Source History
1 package ecommerceSystem;
2 class Order {
3     private int customerId;
4     private int orderId;
5     private Product products[];
6     float totalPrice;
7     public Order(int customerId, int orderId, Product[] products, float totalPrice) {
8         this.customerId = customerId;
9         this.orderId = orderId;
10        this.products = products;
11        this.totalPrice = totalPrice;
12    }
13    public void setCustomerId(int customerId) {
14        customerId = Math.abs(s: customerId);
15        this.customerId = customerId;
16    }
17    public void setOrderId(int orderId) {
18        orderId = Math.abs(s: orderId);
19        this.orderId = orderId;
20    }
21    public void setTotalPrice(float totalPrice) {
22        totalPrice = Math.abs(s: totalPrice);
23        this.totalPrice = totalPrice;
24    }
25    public int getCustomerId() {
26        return customerId;
27    }
28    public int getOrderId() {
29        return orderId;
30    }
31 }
Output 19:32 INS Windows (CRLF)
0:31 24°C 7:47 PM 4/22/2024
```

The screenshot shows the NetBeans IDE with the `Order.java` file open. The code defines methods for setting and getting total price, customer ID, and order ID, as well as a method to print order information. The IDE interface includes a menu bar, a toolbar, and a sidebar with project and service views.

```
21 public void setTotalPrice(float totalPrice) {
22     totalPrice = Math.abs(totalPrice);
23     this.totalPrice = totalPrice;
24 }
25 public int getCustomerId() {
26     return customerId;
27 }
28 public int getOrderId() {
29     return orderId;
30 }
31
32 public float getTotalPrice() {
33     return totalPrice;
34 }
35 public void printOrderInfo() {
36     System.out.println("Here's your order's summary: ");
37     System.out.println("Order ID: " + orderId);
38     System.out.println("Customer ID: " + customerId);
39     System.out.println("Products: ");
40     for (Product p : products) {
41         System.out.println(p.getName() + " - $" + p.getPrice());
42     }
43     System.out.println("Total price: " + totalPrice);
44 }
45 }
46
```

The screenshot shows the NetBeans IDE with the `Output` window open, displaying the execution of the `ECommerceSystem` application. The output shows the user interacting with the system, entering their ID, name, and address, and adding products to their cart. The final output shows the order summary and the total price.

```
run:
Welcome to the E-Commerce system!
Please enter your id
23011613
Please enter your name
Yassmin
Please enter your address
Alexandria
How many products do you want to add to your cart?
4
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
Which product would you like to add? 1- Smartphone 2- T-Shirt 3- OOP
1
Your total is: 679.87Would you like to place the order? 1-Yes 2-No
1
Here's your order's summary:
Order ID: 1
Customer ID: 23011613
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
smartphone - $599.9
Total price: 679.87
BUILD SUCCESSFUL (total time: 16 seconds)
```