```
Class Product (
    private int productId;
   private String name;
   private double price;
   public Product(int productId, String name, double price) (
        this.productId = Math.abs(productId);
        this.name = name;
        this.price = Math.abs(price);
   public int getProductId() {
        return productId;
    public void setProductId(int productId) {
        this.productId = Math.abs(productId);
    public String getName() [
        return name;
    public void setName(String name) {
        this.name = name;
   public double getPrice() {
```



```
public void setPrice (double price) {
       this.price = Math.abs(price);
class ElectronicProduct extends Product {
   private String brand;
   private int warrantyPeriod;
   public ElectronicProduct(int productId, String name, double price, String brand, int warrantyPeriod) {
       super(productId, name, price);
       this.brand = brand;
       this.warrantyPeriod = Math.abs(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(warrantyPeriod);
```



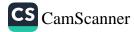
```
private String size;
 79
 80
           private String fabric;
 81
           public ClothingProduct(int productId, String name, double price, String size, String fabric) {
 82 -
               super(productId, name, price);
 83
               this.size = size;
 84
               this.fabric = fabric;
 85
           public String getSize() {
 87
 88
               return size;
 89
           public void setSize(String size) {
 90
               this.size = size;
 91
 92
           public String getFabric() {
 93
               return fabric;
 94
 95
           public void setFabric (String fabric) [
 96
               this.fabric = fabric;
 97
 98
99
       class BookProduct extends Product {
100
           private String author;
101
           private String publisher;
102
103
           public BookProduct(int productId, String name, double price, String author, String publisher) {
104
105
               super(productId, name, price);
               this.author = author;
106
```



```
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;
class Customer (
   private int customerId;
   private String name;
   private String address;
   public Customer(int customerId, String name, String address) [
       this.customerId = Math.abs(customerId);
       this.name = name;
       this.address = address;
   public int getCustomerId() {
       return customerId;
```



```
public void setCustomerId(int customerId) (
        this.customerId = Math.abs(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;
class Cart (
    private int customerId;
    private int nProducts;
    private Product[] products;
    public Cart(int customerId, int nProducts) {
        this.customerId = Math.abs(customerId);
        this.nProducts = Math.abs(nProducts);
        this.products = new Product[nProducts];
```



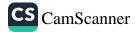
```
public int getCustomerId() {
    return customerId;
public void setCustomerId(int customerId) {
    this.customerId = Math.abs(customerId);
public int getnProducts() {
    return nProducts;
public void setnProducts(int nProducts) {
    this.nProducts = Math.abs(nProducts);
public void addProduct(int index, Product product) {
    if (index >= 0 && index < nProducts) {
        products[index] = product;
    } else {
        System.out.println("Invalid index to add product.");
public void removeProduct(int index) {
    if (index >= 0 && index < nProducts) (
        products[index] = null;
    } else {
        System.out.println("Invalid index to remove product.");
```



```
public double calculatePrice() {
         double totalPrice = 0;
         for (Product product : products) {
             if (product != null) {
                 totalPrice += product.getPrice();
         return totalPrice;
    public void placeOrder() {
    Product[] getProducts() {
        throw new UnsupportedOperationException("Not supported yet.");
class Order [
   private final int customerId;
   private final int orderId;
   private final Product[] products;
   private final double totalPrice;
   public Order(int customerId, int orderId, Product[] products) {
       this.customerId = Math.abs(customerId);
       this.orderId = Math.abs(orderId);
       this.products = products;
       this.totalPrice = calculateTotalPrice();
```



```
ClothingProduct clothingProduct = new ClothingProduct(2, "T-shirt", 19.99, "Medium", "Cotton");
BookProduct bookProduct = new BookProduct(3, "OOP", 39.99, "O'Reilly", "X Publications");
System.out.println("Enter your customer ID:");
int customerId = scanner.nextInt();
scanner.nextLine(); // Consume newline
System.out.println("Enter your name:");
String name = scanner.nextLine();
System.out.println("Enter your address:");
String address = scanner.nextLine();
Customer customer = new Customer(customerId, name, address);
System.out.println("How many products do you want to order?");
int nProducts = scanner.nextInt();
Cart cart = new Cart(customerId, nProducts);
for (int i = 0; i < nProducts; i++) [
    System.out.println("Enter product type (1 for Electronic, 2 for Clothing, 3 for Book):");
    int productType = scanner.nextInt();
    scanner.nextLine(); // Consume newline
    switch (productType) {
        case 1:
            cart.addProduct(i, electronicProduct);
            break:
        case 2:
            cart.addProduct(i, clothingProduct);
            break;
```



```
break;
        case 3:
            cart.addProduct(i, bookProduct);
           break;
        default:
            System.out.println("Invalid product type.");
System.out.println("Do you want to place an order for the products in the cart? (yes/no)");
String choice = scanner.next();
if (choice.equalsIgnoreCase("yes")) {
    cart.placeOrder();
   Order order = new Order(customerId, 123456, cart.getProducts());
    order.printOrderInfo();
) else (
    System.out.println("Order not placed.");
```



```
Output - eCommerce (run) ×
     Welcome to the E-Commerce System!
     Enter your ID:
     222110
     Enter your name:
     2020
     Enter your adress:
     mandara
     How many products you want to add to your cart?
     Wich product would you like to add? 1- smartphone 2- T-shirt 3- DOP
     smartphone Added to Cart!
     Wich product would you like to add? 1- smartphone 2- T-shirt 3- OOF
     smartphone Added to Cart!
     Wich product would you like to add7 1- smartphone 2- T-shirt 3- OOP
     OOP Added to Cart!
     Wich product would you like to add? 1- smartphone 2- T-shirt 3- OOP
     T-shirt Added to Cart!
     Your Total is $1259.78
     Would you like to place an order? 1- Yes 2- No
     Here's your order's summary:
     Order ID: 1
     Customer ID: 222110
     Products:
     smartphone - $599.9
     smartphone - $599.9
     OOP - $39.99
     T-shirt - $19.99
     Total Price: $1259.78
     BUILD SUCCESSFUL (total time: 33 seconds)
```

