

Project 1

The project is a simple E-Commerce system that models different types of products and their interactions with customers. The project consists of the following classes:

1. **Product**: This is the superclass that represents a generic product.

Attributes:

- productId: positive integer (else take absolute value)
- name: String
- price: positive float (else take absolute value)

Methods: Setters and getters for the attributes.

2. **ElectronicProduct**: This is a subclass of Product that represents electronic products.

Attributes:

- brand: String
- warrantyPeriod: positive integer (else take absolute value)

Methods: Setters and getters for the attributes.

3. **ClothingProduct**: This is a subclass of Product that represents clothing products.

Attributes:

- size: String
- fabric: String

Methods: Setters and getters for the attributes.

4. **BookProduct:** This is a subclass of Product that represents book products.

Attributes:

- author: String
- publisher: String

Methods: Setters and getters for the attributes.

5. **Customer:** This class represents a customer.

Attributes:

- customerId: positive integer (else take absolute value)
- name: String
- address: String

Methods: Setters and getters for the attributes.

6. **Cart:** This class represents a shopping cart.

Attributes:

- customerId: positive integer (else take absolute value)
- nProducts: positive integer which represents the number of products (else take absolute value)
- products: an array of Product objects of size nProducts.

Methods: Setters and getters for the attributes, addProduct, removeProduct, calculatePrice and placeOrder.

7. **Order:** This class represents an order placed by a customer.

Attributes:

- customerId: positive integer (else take absolute value)
- orderId: positive integer (else take absolute value)
- products: an array of Product objects of size nProducts.
- totalPrice: positive float (else take absolute value)

Methods: printOrderInfo.

8. **EcommerceSystem:** This class serves as the entry point for the project. It contains the main method and demonstrates the usage of the other classes. You'll have to create the products but take orders from the user.

Test case:

1. Create an electronic product with these specifications:
 - name: "smartphone"
 - id: 1
 - price: \$599.9
 - brand: "Samsung"
 - warrantyPeriod: 1
2. Create a clothing product with these specifications:
 - name: "T-shirt"
 - id: 2
 - price: \$19.99
 - size: "Medium"
 - fabric: "Cotton"
3. Create a book product with these specifications:
 - name: "OOP"
 - id: 3
 - price: \$39.99
 - author: "O'Reilly"
 - publisher: "X Publications"
4. Create a customer with **your data**: (take it as input from the user)
 - customerId: your id
 - name: your name
 - address: "address"
5. Create a shopping cart for the customer you created and ask them how many products they want to order and what they are and add them in the cart.
6. Ask the user if they want to place an order for the products in the cart.
7. Print order info.

The output should be:

```
Welcome to the E-Commerce System!
Please enter your id
20231
Please enter your name
Student Name
Please enter your address
address
How many products 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 $679.95996. Would you like to place the order? 1- Yes 2- No
1
Here's your order's summary:
Order ID: 1
Customer ID: 20231
Products:
T-Shirt - $19.99
OOP - $39.99
T-Shirt - $19.99
Smartphone - $599.99
Total Price: $679.95996
```

Bonus:

Create a GUI for this system.

Deliverables:

- Java project
- A pdf containing screenshots of all 8 classes + test case output
- Note: make sure to use your student id and name in the test case