## E-Commerce Platform

An online shopping website manages Customers, Products, and Orders. Customers can purchase multiple products, and an order contains multiple products. Tasks:

Create a base class User with name and email.

Create a Customer subclass with shoppingCart as an attribute.

Create a Product class with productID, name, and price.

Create an Order class that holds multiple Product objects.

Implement associations (Customer has multiple Orders).

Create a hierarchy where Product is the base class and Electronics, Clothing, and Furniture are subclasses. Further extend Electronics into Mobile and Laptop.

Introduce DigitalProduct as another subclass of Product, which is extended by EBook and OnlineCourse.

- Call super in Electronics to set product attributes inherited from Product.
- Use super.methodName() to reuse a method in Clothing.
- Mark final a method generateInvoice() to prevent modifications in subclasses.
- Declare final class OrderIDGenerator that ensures unique order IDs.

## **University Management System**

The system consists of multiple packages:

- 1. `university` (Main package)
- 2. `university.students` (Handles student details)
- 3. `university.faculty` (Handles faculty details)
- 4. `external` (For external organizations collaborating with the university)

Each class in these packages needs to access \*\*Student\*\* details with different access levels (Private, Protected, Public, and No Modifier). Design a Java program that demonstrates \*\*access control\*\* for a `Student` class, considering different \*\*visibility rules\*\* across the following:

Same Class (Inside `Student` class itself)

Same Package Subclass (A subclass inside the same package as `Student`) Same Package Non-Subclass (Another class in the same package but not a subclass)

Different Package Subclass (A subclass in a different package)

Different Package Non-Subclass (A non-subclass in a different package)

- 1. Create a `Student` class inside `university.students` with the following:
  - Private variable: `private int studentID`
  - No Modifier variable: `String studentName`
  - Protected variable: `protected double GPA`
  - Public variable: `public String department`
- 2. Access `Student` members from:
  - Another class in the same package (both subclass and non-subclass).
  - A subclass in a different package.
  - A non-subclass in a different package.
- Test access levels for each variable type (`private`, `protected`, `public`, and no modifier).

## **Online Food Ordering System**

An online food ordering system needs to handle multiple customers placing orders at the same time. The system should ensure that: Users can place orders concurrently, but food preparation follows a first-come, first-served basis, If the restaurant has limited resources (e.g., limited chefs preparing food), the system should handle order processing accordingly, Orders should be processed sequentially even if multiple users place them at the same time.

- Design a Java program that simulates an online food ordering system using multithreading.
- Implement a shared resource (FoodOrderSystem) that maintains a queue of food orders.
- Use synchronized methods or locks to ensure orders are processed one at a time.
- If the kitchen is busy, new orders should wait until the previous order is processed.
- 5. Demonstrate the functionality with at least three user threads placing orders concurrently.