Web Technologies Lab

Lab 03 **Marks 100**

Instructions

Work on this lab individually.

You are **NOT** allowed to use the internet, or mobile phone.

You are **NOT** allowed to borrow anything from your peer student.

What you have to do

Program the following tasks. The name of your files will be according to the task given in this lab.

Task 1 [100]

A retail company wants to implement a Product Management System to categorize different types of products in its inventory. Each product has unique attributes and price rules. The system should allow users to enter product details, apply category-based discounts, and validate data to ensure correctness.

The system must also handle invalid input cases, such as negative prices, out-of-stock products, and expired groceries.

Instruction:

1. Create an interface `Product` with the following methods:
   - double calculatePrice()
   - String getDetails()

2. Define a `BaseProduct` class that implements `Product`
   - with attributes:
     i. String name
     ii. double price
     iii. int stock
   - Also provide:
     i. A constructor to initialize name, price, and stock.
     ii. An overridden getDetails() method to display product details.
     iii. A method to check stock availability before purchase.

3. Implement the following product types:
   - **`Electronics`**
     i. Includes an additional warranty period (in months).
     ii. If the price is above $1000, apply a 10% discount.
   - **`Clothing`**
     i. Includes size (S, M, L, XL) and material (e.g., Cotton, Polyester).
     ii. If stock < 5, apply a 5% discount.
   - **`Groceries`**
     i. Includes expiration date (check if expired before adding).
     ii. If the customer buys more than 5 items, apply a bulk discount of 8%.
   - **`Books`**
     i. Includes author name.
     ii. If price > $50, apply a 7% discount.

4. Implement Exception Handling Using Try-Catch Blocks:
   - Use **try-catch blocks** in constructors to validate product details
   - Use a **finally block** to display `"Product processing complete."` after each attempt.

5. Implement the **`Driver`** class
   - Use Scanner to take user input for product details.
   - Ask the user to select a product type.
   - Create an instance of the selected product type and display its details.
   - Validate stock availability before adding a product.
   - Display product details and final price after applying category-based discounts.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺