LAB 211 Assignment

Type:	Long Assignment
Code:	J1.L.P0024
LOC:	400+
Slot(s):	N/A

Title

Bike Stores Management System – Read and Write File.

Background

ABC Company want to develop a Java console application to perform CRUD operations on Product entities. The application will interact with **three text files** (***Product.txt, Brand.txt, Category.txt***) to manage and persist data. The primary focus is on ensuring data integrity, especially during creation, updating, and deletion operations.

Program Specifications

The **main screen** allows user to select the following functions:

1. Add a product.
2. Search product by product name, return a list of all products that same name.
3. Update product.
4. Delete product.
5. Save products to file.
6. Print list products from the file.

Others - Quit.

Features:

This system contains the following functions:

- **Function 1: Create a Product – 100 LOC**
    - The **Product** has fields: *id, name, brand id, category id, model year, and list price.*
    - Check the valid data with the following conditions:
        - **id**: Generate a unique.
        - **Name**: Ensure name is not empty.
        - **Brand id**: Validate brand id exists in Brand.txt.
        - **Category id**: Validate category id exists in Category.txt.
        - **Model year**: Ensure model year is a valid year.
        - **List price**: Validate list price is a positive number.
    - Create new a **Product**.
    - Ask to go back to the main menu.

- **Function 2: Search Product information by name – 50 LOC**
  - Require to enter a search string (**a part of product name**), and return a list of Products information containing the search string.
  - If the **list Product is null**, the notification "Have no any Product", **else** print the list Product information order by the Model Year ascending.
  - Ask to go back to the main menu.
- **Function 3: Update Product information – 80 LOC**
  - The user enters the **id** from the keyboard.
  - If it does **NOT EXIST**, the notification "Product does not exist". **Otherwise**, the Product can be edited the remaining information:
    - Input Validation: Same validations as in the create function.
    - If Information is blank, then do not change old information.
  - Show the result of the update: **success or failure**.
  - Ask to go back to the main menu.
- **Function 4: Delete Product information – 20 LOC**
  - The user enters the **id** from the keyboard
  - If it does **NOT EXIST**, the notification " Product does not exist". **Otherwise**, the Product can be deleted
  - Must show confirm message
  - Show the result of the delete: success or fail
  - Ask to go back to the main menu.
- **Function 5: Save to file - 50 LOC**
  - Write a list of the Product's information to the file (**Product.txt**).
  - File Structure: **id, name, brand_id, category_id, model_year, list_price**
  - Ask to go back to the main menu.
- **Function 6: Print all lists from file – 100 LOC**
  - **Loading** list Product information from the file into **Collection**
  - Displaying list Product information **order by list price descending**. If the same price, **then order by name ascending**
  - Output format: **id, name, brand_name, category_name, model_year, list_price**
  - Ask to go back to the main menu.

- **Bonus 50 LOC (maximum 450 LOC) if the student applies one of the Design Patterns (such as DAO pattern, Factory pattern, Repository pattern, and so on) in this project. More references for the design pattern:** https://www.tutorialspoint.com/design_pattern/index.htm
- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once on the first slot of the assignment.