

LAB101Assignment

Type:	Long Assignment
Code:	J1.L.P0003
LOC:	400++
Slot(s):	N/A

Title

Manage product – Read and Write File.

Background

In this assignment you are to write a program that will maintain a product management using collection and file.

Program Specifications

Your program should first ask the user the name of the text file that contains the category information and product information. Then, it should open the file and read its contents into a collection of item structures. After the file has opened and the contents had read into the collection, close the file and display the following menu:

1. Add new category
2. Update category
 - 2.1 Update category
 - 2.2 Delete category
3. Add new product
4. Update product
 - 4.1 Update product
 - 4.2 Delete category
5. Order product
6. Show order list report

Others- Quit Program

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

▪ **Function 1: Add new category - 50 LOC**

- This function requires user input a category information include category id and category name.
- The new category must be appended to the end of the structure (in memory). Do not allow duplicate category (names nor id) in the storage.
- System allows adding the category to collection.
- Application asks user to go back to the main menu.

▪ **Function 2: Update category**

▪ **Function 2.1: Update category information – 50 LOC**

- This function requests user entering the category id.
- If it does not exist, the notification “Category does not exist” is shown. Otherwise, user can edit of the in category remaining information. If information is blank, then system does not change old information.
- System shows the result of the update action (success or fail).

- Application asks user to go back to the main menu.
- **Function 2.2: Delete category information – 50 LOC**
 - This functions requests user entering the category id.
 - If it does not exist, the notification “Category does not exist” is shown. Otherwise, user can delete the category.
 - System shows the result of the delete action (success or fail).
 - Application asks user to go back to the main menu.
- **Function 3: Add new product – 50 LOC**
 - This function requests user inputting a product information include product id, product name, price, quantity, category id.
 - System checks the valid data with the following conditions:
 - Product id is not be allowed to duplicate in the database.
 - Price must be a positive number
 - Quantity must be a positive integer number
 - Category id must be in category list
 - System allows adding product to collection
 - Application asks user to go back to the main menu.
- **Function 4: Update product information**
 - **Function 4.1: Update product information – 50 LOC**
 - This function requests user entering the product id.
 - If it does not exist, the notification “Product does not exist” is shown. Otherwise, user can edit of the remaining information. If information is blank, then system does not change old information.
 - System shows the result of the update action (success or fail).
 - Application asks user to go back to the main menu.
 - **Function 4.2: Delete product information – 50 LOC**
 - This functions request user entering the product id.
 - If it does not exist, the notification “Product does not exist” is shown. Otherwise, user can delete the product.
 - System shows the result of the delete action (success or fail).
 - Application asks user to go back to the main menu.
- **Function 5: Order product**
 - **Function 5.1: Display all products – 50 LOC**
 - This function loads product list information from the file into Collection.
 - The application displays product list information on the screen. For example:

Products listing:

++ No ++	++ Product name ++	++ Price ++
1	Coconut	2.5 \$
2	Coffee	3.5 \$
3	Candy	4.0 \$
4	Sugar	3.5 \$

- Application asks user to go back to the main menu.

- **Function 5.2: Order product – 50 LOC**

- To order, customer selects a product, for example: when customer selects No 1, the program shows:

You selected: Coconut

Please input quantity:

- After customer inputs quantity of product, the program shows message: Do you want to order now (Y/N). If customer selects N, the program returns to product list to continue ordering. If select Y, the program displays:

```
|++Product++|++Quantity++|++ Price++|++ Amount++|
Coconut      3      2.5 $      7.5 $
Total: 7.5 $
Input your name:
```

- Customer inputs his/her name to finish ordering.
- Order detail will be add in to the order list, who buy how many products.
- Ask to go back to the main menu.

- **Function 6: View order list - 50 LOC**

- Print all order detail in order list on the screen. For example:

```
Customer: Marry Le
Product | Quantity | Price | Amount
1. Apple   3      1.5 $   4.5 $
2. Candy   2      2.5 $    5 $
Total: 9.5 $
```

```
-----
Customer: Tony Dung
Product | Quantity | Price | Amount
1. Sugar   3      3.5 $  11.5 $
2. Coffee   2      3.5 $    7 $
Total: 13 $
```

- Application asks user to go back to the main menu.
- **Function 7: Read file - 50 LOC**
 - This function loads product list from file product.txt, list category from category.txt in to Collection
 - System writes product list, category list from program to file product.txt and category.txt.

- **Function 8: R Show order list report - 100 LOC (Ext)**
 - This function allows printing inventory reports of the store.
- 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.