LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0001**
LOC:	**500**
Slot(s):	**N/A**

Title

Order Management

Background

A glucosamine store needs an order management program with basic requirements such as creating orders, displaying order information, updating information, etc. All of the information (products, customers, orders) has been taken from the text files products.txt, customes.txt, and orders.txt as follows:

File products.txt	Description
P001,Move Free,box of 200 tablets,Schiff (USA),658.000 P002,Glucosamine,box of 150 tablets,Schiff (USA),490.000 P003,Glucosamine,box of 375 tablets,Kirkland (USA),583.000 P004,Fish Oil,box of 400 tablets,Kirkland (USA),389.000 P005,Glucosamine,box of 240 tablets,Puritan's Pride (USA),494.000 P006,Glucosamine,box of 180 tablets,Blackmores (Australia),539.000 P007,Glucosamine,box of 360 tablets,Orihiro (Japan),299.000 P008,Glucosamine & Fish Oil,box of 90 capsules,Blackmores (Australia),430.000	**Information in a line**: <**productID**, productName, unit, origin, price>

File customers.txt	Description
C001,NGUYEN THI BE,TAN BINH,0918457895 C002,LE HOANG NAM,BINH THANH,0988878987 C003,TRAN THI CHIEU,QUAN 9,0903798798 C004,MAI THI QUE ANH,QUAN 10,0919333575 C005,LE VAN SANG,BINH TAN,0989333125 C006,TRAN HOANG KHAI,QUAN 3,0913069768	**Information in a line**: <**customerID**, customerName, customerAddress, customerPhone >

File orders.txt	Description
D006,C004,P001,2,12/14/2022,false D001,C002,P008,5,11/15/2022,true D002,C006,P004,3,11/20/2022,true D004,C001,P003,4,12/5/2022,true D003,C002,P007,6,12/3/2022,true D005,C003,P005,3,12/13/2022,false	**Information in a line:** <**orderID**, **customerID**, **productID**, orderQuantity, orderDate, status>

Program Specifications

Build a Order Management program. With the following basic functions

1. List all Products
2. List all Customers
3. Search a Customer based on his/her ID
4. Add a Customer
5. Update a Customer
6. Save Customers to the file, named customers.txt
7. List all Orders in ascending order of Customer name
8. List all pending Orders

9. Add an Order
10. Update an Order
    10.1. Update an Order based on its ID
    10.2. Delete an Order based on its ID
11. Save Orders to file, named orders.txt

Others- Quit

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.

Function details

This system contains the following functions:
The application should display a main menu and ask users to select an option.

- **Function 1: Print all Products – 25 LOC**
  - The program will print a list of all products in default ordering from the products.txt file.
  - The program must allow the user to return to the main menu.
- **Function 2: Print all Customer Information – 25 LOC**
  - The program will print a list of all customers in default ordering from the customers.txt file.
  - The program must allow the user to return to the main menu.
- **Function 3: Search a customer by his/her ID – 50 LOC**
  - The user inputs the customer code that he/she wants to search information.
  - The program shows the customer information that is matched the search string.
  - If no customer is existed, the screen shows "This customer does not exist" message. Otherwise, the user information will be shown.
  - The program must allow the user to return to the main menu.
- **Function 4: Add new Customer – 50 LOC**
  - The application requires inputting a customer's information with customer's ID, customer's Name, customer's Address, and customer's Phone number fields.
  - The system will check the data validation with the following conditions:
    - The customer's ID field is not allowed to duplicate in the database.
    - The customer's Name, customer's Address, and customer's Phone number fields do not allow to contain null value.
    - The customer's Phone is number string which has length from 10 to 12 characters.
    - …
  - All submitted information will be stored into collection type.
  - The program should ask the user to choose creating new customer continuously or going back to the main menu.
- **Function 5: Update Customer – 50 LOC**
  - The application requires inputting the customer's id.
  - If customer does not exist, the notification "Customer does not exist" message is shown. Otherwise, user can edit of the customer's information.
    - If information is not inputted, the old information will not be changed.
  - The system should show the result of the update action with success or fail status.
  - The program must allow the user to return to the main menu.

- **Function 6: Save Customers to the file – 50 LOC**
  - The program will save all information of the customer in the store to the customers.txt file.
  - The system should show the result of this action with success or fail status.
  - The program must allow the user to return to the main menu.
- **Function 7: Print all Orders – 25 LOC**
  - The program will print a list of all orders in ascending order of customer name from the orders.txt file.
  - The program must allow the user to return to the main menu.
- **Function 8: Print all pending Orders – 25 LOC**
  - The program will print a list of all pending orders from the orders.txt file.
  - The program must allow the user to return to the main menu.
- **Function 9: Add new Orders – 50 LOC**
  - The application requires inputting Order's ID information. The Order's ID value is unique in storage.
  - The submenu should be shown to the user choosing Customers
  - The submenu for Products should be same.
  - The user is required to input order's Quantity and order's Date fields
    - Notes: the order's Date cannot be blank.
  - The user is required to input status (default status field is false value)
  - The system will create a new order, add a new order to the collection.
  - The program should ask the user creating new order continuously or going back to the main menu.
- **Function 10: Update Order – 100 LOC**
  - Student should create a submenu **Update** to update or delete an Order based on its ID.
    - **Function 10.1: Update order information – 50 LOC**
      - The application requires inputting the order's ID value.
      - If Order does not exist, the notification "Order does not exist" message is shown. Otherwise, user can edit of the order's status.
        - If information is blank, the old information will not be changed.
      - The system should show the result of this action with success or fail status.
      - The program allows to ask user to go back to the main menu.
    - **Function 10.2: Delete order – 50 LOC**
      - The application requires inputting the order's ID value.
      - If Order does not exist, the notification "Order does not exist" message is shown. Otherwise, the order will be deleted.
        - The system must show the confirm message before delete action is taken.
      - The system should show the result of this action with success or fail status.
      - The program allows to ask user to go back to the main menu.
- **Function 11: Save Orders to the file – 50 LOC**
  - The program will save all information of the order in the store to the orders.txt file.
  - The system should show the result of this action with success or fail status.
  - The program allows to ask user to go back to the main menu.

Guidelines

- The above specifications are only basic information. The requirements of the assignment can be more flexible depending on your instructor.
- The instructor will explain the requirement on the first slot of the assignment.