|  |  |  |
| --- | --- | --- |
| **LAB 211 Assignment** | **Type:** | **Long Assignment** |
| **Code:** | **01** |
| **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. Where the input data (products, customers, orders) 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
    1. Update an Order based on its ID
    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:**

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 order from the file products.txt.
* The program must allow the user to return to the main menu.
* **Function 2: Print all Customes - 25 LOC**
* The program will print a list of all customers in default order from the file customers.txt.
* The program must allow the user to return to the main menu.
* **Function 3: Search a Customer by his/her ID - 50 LOC**
* User inputs the customer code want to search.
* The program shows the customer information that has customer code that matches the search string.
* If no customer is existed, the screen shows message “This customer does not exist”.
* The program must allow the user to return to the main menu.
* **Function 4: Add new Customer - 50 LOC**
* Require to input a customer’s information: customerID, customerName, customerAddress, customerPhone.
* Check the valid data with the following conditions:
* customerID is not allowed to duplicate in the database.
* customerName, customerAddress, customerPhone not allow is null.
* customerPhone is number string which has length from 10 to 12.
* …
* Add the customer to collection.
* Ask to continuous create new customer or go back to the main menu.
* **Function 5: Update Customer - 50 LOC**
  + Require enter the customer id.
  + If customer does not exist, the notification “Customer does not exist”. Otherwise, user can edit of the in customer’s information. If information is blank, then not change old information.
  + Show the result of the update: success or fail.
  + 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 file customers.txt.
  + Process of this function should be shown 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 file orders.txt.
* 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 file orders.txt.
* The program must allow the user to return to the main menu.
* **Function 9: Add new Orders - 50 LOC**
* Require to input OrderID, OrderID must be not duplicated.
* Create a submenu to choose Customers and another submenu to choose Products.
* Require to input orderQuantity, orderDate: can not be blank.
* Require to input status, default status is false.
* Create a new order with inputted data;
* Add a new order to the collection.
* Ask to continuous create new customer or go back to the main menu.
* **Function 10: Update Order - 100 LOC**
* Create a submenu **Update** to update or delete an Order based on its ID.
* **Function 10.1: Update order information - 50 LOC**
  + Require to input the orderID.
  + If Order does not exist, the notification “Order does not exist”. Otherwise, user can edit of the order’s status. If information is blank, then not change old information.
  + Show the result of the update: success or fail.
  + Ask to go back to the submenu Update.
* **Function 10.2: Delete order - 50 LOC**
  + Require to input the orderID.
  + If Order does not exist, the notification “Order does not exist”. Otherwise, user can delete the order.
  + Must show the confirm message before delete.
  + Show the result of the delete: success or fail.
  + Ask to continuous or go back to the submenu Update.
* The program must allow the user to return 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 file orders.txt.
* Process of this function should be shown success or fail status.
* The program must allow the user to return 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.