LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0013**
LOC:	**500**
Slot(s):	**N/A**

Title

The Vehicle Management

Background

Mr. QuanMX plans to develop a program to manage the vehicles in his show room. Vehicles have many properties in common. This management program needs to have basic functions such as: add, edit, delete, search. The program must be designed so that adding a new vehicle is easy. Let's build your idea based OOP model.

Program Specifications

Build a management program. With the following basic functions

0. Build your data structure
1. Load data from file
2. Add new vehicle
3. Update vehicle by ID
4. Delete vehicle ID
5. Search vehicle
   5.1 Search by name(descending)
   5.2 Search by id
6. Show vehicle list
   6.1 Show all
   6.2 Show all (descending by price)
7. Store data to file

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. All vehicle's information in the show room is contained in file vehicles.txt.

In the current stage, the show room has 2 groups of vehicle as follows:

- Car: have the properties such as id, name, color, price, brand, type (sport, travel, etc.), year of manufacture.
- Motorbike: have the properties such id, name, color, price, brand, speed, require license or not require. In this group has especial function is makeSound which print out the message “Tin tin tin".

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

- **Function 0: Build the data structure - 50 LOC**
  - Classes, abstract classes, Interfaces.
  - Use only one collection to store animals.
- **Function 1: Load data from file - 50 LOC**
  - Load all data in the file into the collection.
- **Function 2: Add new vehicle - 50 LOC**
  - Create a submenu that allows the user to add vehicle to the show room.
  - Remember that the constraints must be checked
  - Add the new vehicle to collection.

    - Ask to continuous create new vehicle or go back to the main menu.
  - **Function 3: Update vehicle – 50 LOC**
    - Require enter the vehicle's id.
    - If vehicle does not exist, the notification "Vehicle does not exist". Otherwise, user can start input new information of vehicle and update.
    - If new information is blank, then not change old information.
    - Remember that the constraints must be checked
    - Then system must print out the result of the updating.
    - After updating, the program returns to the main screen.

- **Function 4: Delete vehicle – 50 LOC**
    - User can delete any vehicle in the showroom by id.
    - Before the delete system must show confirm message.
    - Show the result of the delete: success or fail.
    - After delete, the program returns to the main screen
- **Function 5: Search vehicle**
    - Create a submenu that allows the user to select way to search: search by name or by id.
  - **F.5.1: Search by Name – 50 LOC**
    - User input the text want to search.
    - The system will search in the show room, and return all vehicle that has name contain the search string.
    - Show result list: all information of vehicle(descending).
  - **F.5.1: Search by id – 50 LOC**
    - The user enters the vehicle id.
    - The system searches the show room, and returns the vehicle that has id same with the search string.
    - Show result: all information of vehicle.
- **Function 6: Show vehicle list**
    - Create a submenu that allows the user to select way to show: show all or by type.
  - **F.6.1: Show all – 50 LOC**
    - The system will show this list of vehicle in the show room.
    - Show result list: all information of vehicle.
  - **F.6.2: Show all (descending by price)– 50 LOC**
    - The system will show this list of vehicle in the show room.
    - Show result list: all information of vehicle (descending by price)
    - If vehicle is a motorbike type then call the makeSound function.

- **Function 7: Store data to file-50LOC**
    - Store data in collection to file.

- 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.
- All errors must be handled, not accepted interrupt the program.