Fall 2022 PROG1925 – Programming Concepts I
Assignment 4

[Maximum points: 30]

In this part of the assignment, you are required to create a C# Console Application project. This, again, is a team assignment. Please enroll yourselves into a team. It will automatically put you in a team at the end of the day.

This assignment is focused on you demonstrating the use of arrays and test plans

Create a C# console program that creates and records information for a used car dealership. To manage the model of inventory, the owners only want to store data for a total of 12 cars. Assume the store can only hold three car brands with ONLY one model each from Sedan (1), Hatchback (2), SUV (3), or Pickup truck (4).

e.g Honda brand can not have two Sedan(1)s, but can have up to one each from different models

When the program is first to run, get the name of each of the three brands and store in an (string) array called "brands". Use a for loop to perform this operation. Then create another (string) Array called "inventory" to store car inventory. This array will store each car as "brand-model" format. E.g: Mazda-4 for Mazda pickup.

Afterwards, the following information must be displayed in a menu:

A. Add new Car details -Brand Name & Model (for example "Ford SUV")
B. Edit existing Car details (Brand Name & Model)
C. Display all Cars in store (display the Brand Name & Model)
D. Delete Car Information
E. Exit the program

After a choice from the main menu is selected, the program must accurately perform the necessary operations and returns to the main menu unless option "E" is selected.

<u>Note the following for each option</u>

A – Add new Car details

- For option "A" request the Brand Name and the Model using 2 separate requests.
    - The Brand Name –must be one of the three brand names entered at the start of the program
    - The Model - must be either number 1,2,3 or 4
- You are responsible for putting the final format together which must be Brand-Model example "Ford-1".
- The program must display "Error Car record already exists" if one was already entered and previously stored (so this must be checked for before attempting to save.
- You must use error exception handling for full marks
    - throw an exception error if the user enters a number less than 1 or greater than 4.

- In case of an invalid input, the user must be prompted to re-enter the value until a valid value has been entered.
- If the input is correct, save the brand information to the array by searching the array, for the first available record spot (which can be a previously deleted brand ("NONE" – see option D below) OR a new record – once there are less than 20 Cars stored.
- notify the user that the record has been saved and repeat the process for option A – user should be able to enter new brand information until they enter "DONE" for the brand name field or when the array has 20 records. Use a while loop to perform this task.

B - Edit existing Car details (Brand Name & Model)

- For option "B" ask the user for the brand entry they need to edit; it must be entered in the format Brand-model (example Ford-2)
- If no record exists, the program must display " Brand record not found for that entry"
- If the record exists, display the message "brand information found" then
- Prompt the user to enter new values ("please enter updated information")
    - For each value as with menu option A, use separate requests for Brand and model and perform the same validations.
- If the information entered is correct, update that specific position in the array (that contained the old record), then notify the user that the record is updated – pressing enter should then display the main menu.

C - Display all Cars in store (display the Brand Name & Model)

- For option "C" the program must display all the saved brand information to the screen.
- After which when the user presses enter the program must return to the main menu – prompt the user to press enter to return to the main menu.

D - Delete existing Car details (Brand Name & Model)

- For option "D" ask the user for the brand entry they need to delete; it must be entered in the format Brand-model (example Ford-2)
- If no record exists, the program must display " Brand record not found for that entry" then return to the main menu
- If the record exists, display the message "brand information found. Are you sure you want to delete Y/N"
    - If the user enters "Y" replace the brand information record with the value "NONE" then return to the main menu.
    - IF the user enters "N" return to the main menu.

Sample Test Plan - 5 marks:

Please pay attention to the marking Rubric (available on the class portal in the same location as an assignment - eConestoga). Each of the **items #2-10 under option A** - represents a test item on your plan and must be represented as shown in the test plan example below:

Outline the test plan in this example format shown below.

Test Plan for Assignment #4 Coder: <your name > Tester: <your name > (in this case)

Test description From Rubric	Expected Result	Test Pass or Fail
<description of what is being tested>	<what you expect to happen when that feature is tested or action happens >	<If the expected result happens pass if it doesn't happen fail>
Example: User Notified when record is saved	**Prompt shown on screen "Record Saved Successfully"**	**Pass**

Adhere to coding and assignment submission standards/guidelines. Marks will be taken off, if these standards/guidelines are not followed. The format for submitting the assignment is as follows:

1. **eConestoga Submission:**
   The **ENTIRE** project folder uploaded to the assignment folder under "Assignment 4".
2. **Demonstration of program in class:**
   Please have your program uploaded to eConestoga and running when you are ready to demonstrate your work.