Part I: PA3 Outline (10 points). Create an outline in comments/psuedocode for the programming assignment below. Place your comments in the appropriate files: main.cpp, functions.h, dealer.cpp, dealer.h, dealer.cpp (as noted below). Place into a file folder named LastnamePA3, the zip the content and hand in a zip file to Canvas.

Part II: PA3: Car Dealership (40 points)

For Programming Assignment 3 you will be creating a program to manage cars in a dealership. This will again be a menu driven system. You will make a vector of Dealers (there can be zero to many dealers). Each dealer can have zero to many cars on the lot, represented not by a vector, but by a dynamic array. The following is your menu:

1. Read Dealers and Cars from file
2. Display Dealers
3. Choose a Dealer Number, Display Cars
4. Choose a Dealer Number, Add Car
5. Choose a Dealer, Sort cars by VIN (EXTRA CREDIT)
6. Write Dealers and Cars to file (EXTRA CREDIT)
7. Exit

Your program will be object oriented (using classes) with the following UML representing the classes (See Chapter 13 for UML Access Specification information):

Dealer

+ Dealer ( ) // Don't forget to add a new car pointer when you use the Dealer default / constructor

+ Dealer (dName:string, dNumber:int)

- dealerName: string

- dealerNumber: int

- carArrayPtr: pointer to a Dynamic Car Array (Note: make sure this is set to nullptr)

- numberOfCars: int

+ setDealerName (_dealerName:string): void

+ setDealerNumber (_dealerNumber:int): void

+setCarArrayPtr (_carArrayPtr: Car *): void //(This is where you use the new)

+setNumberOfCars (int _numberOfCars): void

+ getDealerName ( ): string

+ getDealerNumber ( ): int

+ getCarArrayPtr ( ): Car *

+getNumberOfCars: int

+ friend operator << (out: ostream &, Dealer: _dealer):ostream &
    //Print the Dealer Name and Number and Blank line for a specific dealer.

Car

- VIN:string

- make:string

- model:string

- year:int

- price:double

+ Car( )

+ Car(vVIN:string, vMake:string, vModel:string, vYear:int, vPrice:double)

+ getVIN( ):string

+ getMake( ):string

+ getModel( ):string

+ getYear( ):int

+ getPrice( ):double

+ setVIN(_VIN:string):void

+ setMake(_make:string):void

+ setModel(_model:string):void

+ setYear(_year:int):void

+ setPrice(_price:double):void

+ friend operator << (out: ostream &, Car: _car):ostream &
    //Print the VIN, Make, Model, Year, Price and Blank line for a specific car.

You will have at least four files for your program (Use these file names!): main.cpp, functions.h, Dealer.h, Dealer.cpp. Place into a file folder named LastnamePA3, the zip the content and hand in a zip file to canvas

- **main.cpp**: this will be your driver file.
- **functions.h**: this will contain your functions for each of the menu items. You are welcome to sperate function prototypes from implementation using functions.h and functions.cpp
- **Dealer.h:** this will contain the class declarations for car and dealer including the operator <<
- **Dealer.cpp:** this will contain the class implementations for car and dealer

You will be storing your dealer objects in a **vector**. In the main function, you will create a vector of Dealers with an initial size of zero (0). When the menu are called, you will have to check your vector and ensure that it is not empty before referring to the index number of the vector. Your Dealer vector points to a **dynamic array** of cars. Again, when you access the cars class you should verify that it is not set to nullptr.

You will not have any global variables.

Each menu item will have a corresponding function, and the definition of the function will be found in the file functions.h. All input/output will be done in the functions and not in main (expect asking for what menu option the user wants).

The following are the details for each of your menu options:

1. Read Dealers and Cars from file
2. Display Dealers
3. Choose a Dealer Number, Display Cars
4. Choose a Dealer Number, Add Car
5. Choose a Dealer, Sort cars by VIN (EXTRA CREDIT)
6. Write Dealers and Cars to file (EXTRA CREDIT)
7. Exit

1. Pass the vector of dealers (from main) into the function by reference. Each Dealer will have a name and number followed by the number of cars in the dealership (on separate lines). Then you will read in a Vin, Make, Model, Year and Price for each car (on separate lines). You can assume that the number of cars as well as the Vin, Make, Model, Year and Price will be complete for each car. Each dealer can have Your file format will be as follows:

Dealer Name
Dealer Number
2
Car1 Vin
Car1 Make
Car1 Model
Car1 Year
Car1 Price
Car2 Vin
Car2 Make

Car2 Model
Car2 Year
Car2 Price
Dealer Name
Dealer Number
1
Car1 Vin
Car1 Make
Car1 Model
Car1 Year
Car1 Price

Here is an example file:

John Elway Dodge
1
2
VIN123
Dodge
Ram 1500 Quad Cab
2017
45186.00
VIN456
Chevrolet
Traverse Premier SUV
2017
47868.00
Tesla Cherry Creek
25
1
VIN789
Tesla
Model X
2017
105200

2. Pass the vector of dealers (from main) into the function by reference. Display the Dealer Name and Number for each dealer (put a blank line between dealers). Use the << operator with cout (cout << dealer[x];)

3. Pass the vector of dealers (from main) into the function by reference. Display the Dealer Names and Numbers (menu #2). Ask the user for the dealer number, display all of the cars by using a for loop (from zero to size of the car array) using the << operator with cout (cout << dealer[x].carArrayPtr[x];)

4. Pass the vector of dealers (from main) into the function by reference. a) Display the Dealer Names and Numbers (menu #2). b) Ask the user for the dealer number then the new car information. Follow the steps found in CSCI 1411 Lab 09 to increase the dynamic array. (Essentially you will make a new

array, one bigger than the previous car array. Then you will copy everything over from the old car array to the new one. Then you will point the car pointer to the new array)

You can earn up to 20% **extra credit (58/50)** if 1-4 and 7 work properly and you get 5 and 6 working properly.

5. Pass the vector of dealers (from main) into the function by reference. Display the Dealer Names and Numbers (menu #2).6. Ask the user for the dealer number. Sort the cars by vin and display them (with blank lines between each car

6. Pass the vector of dealers (from main) into the function by reference. Also pass in the filestream (from main) by reference. (Use a different file than your original OR make sure to close the files before overwriting). Write all of the Dealers and Cars to a separate output file in the same format as in item #1

7. Exit out of the program