Assignment 9: C-Strings, String Objects, and Classes

[1] **Objectives**: The main purpose of this assignment is to make sure you are familiar with C-string, String class, and classes in general. The old C-strings are being used a lot in the existing applications, so it is important to know how it works. In addition, it is important that you know how to use functions to structure your program.

[2] **Description**: In this assignment you are going to build a computer application for car dealerships. This application needs to keep track of all the cars the dealership has at the moment. The user of the application should be able display all cars, the cars with prices within a range, and all the cars made by a give company.

   a) Define a class for "Auto", with the following members:
   - make, array of character of size 20,
   - model, a string,
   - year, a 4-digeit number
   - color, a string,
   - price, double,
   - setAll(string make, string model, string color , int year, double price);
   - getPrice(), return the price of the car;
   - getMake() return the make of the car;
   - display() display all the information about the car;

   b) Define another class called "Dealer", with the following members:
   - Cars, An array of Autos (up to 100 in size).
   - noOfCars, The number of cars currently in the dealership.
   - addCar(), to insert a new car to the array
   - getNoOfCars(), return the number of cars stored in the array
   - displayAll(), print all the cars
   - displayBetween(double min, double max), display all the cars with prices between min and max.
   - displayByMake(char mk[]), display all the cars that has make value equal to mk.

Your main menu should look like: (This is the final menu that I used. The sample output used a slightly different menu).

   *********** Class Deealership ******* Total cars = 9**

   1. Display all cars
   2. Display all cars within price range
   3. Display all cars with specific make
   4. Exit

The difficulty to implement this program is the lack of "file input" at this time. So we wrote such a function for your use. A template for this assignment will be provided. You don't have to understand the detail steps of the function (yet) but have to know how to call it. This is probably a good exercise for you to go through such a practice at this stage. Keep in mind that this is my way of getting around the file

input issue. This may not be the best way to do the work once we get to Ch. 12. As much as I hate it, I have to put one variable at the global level to simplify the code. There is no requirement to keep the inventory in any particular order. If you have extra time, you may try to sort them in increasing order of the Make (for your personal satisfaction).

[3] **Input**: Menu-driven interactive. A test data will be stored in a file called "prog9in.txt" for you to test the program. Each car record is stored in a separate line. Within a line, all fields are separated by one or more blanks. The function getRecord() provided will return all five fields one at a time. It will also return true or false depending on whether there is more data in the file. Because you have to define the auto classes, it is not possible for the function to return an auto object.

[4] **Output**: A sample output is shown below. First name and last name should be put together properly.

```
File "prog9in.txt" opened.
    Make      Model      Color     Year      Price
    ---------------------------------------------
  Toyota     Camry        red      2002       4000
    Ford     Taurus      green      2004       3500
  Nissan     Altima       grey      2009       7000
    Ford     Focus       white      2001       2000
mercedes    C-Class      black      2010      20000
  Toyota     Yaris        blue      2007       6500
  Nissan     sentra      white      2014      15000
  Toyota    corolla      green      2007       7000
   Honda     civic       white      2006       6500

********** Class Dealer ******* Total cars = 9
  1. Display all cars
  2. Display all cars within price range
  3. Display all cars with specific make
  4. Exit
>> 2
Enter the minimum amount you want to spend: 4000
Enter the maximum amount you can pay: 7000
    Make      Model      Color     Year      Price
    ---------------------------------------------
  Toyota     Camry        red      2002       4000
  Nissan     Altima       grey      2009       7000
  Toyota     Yaris        blue      2007       6500
  Toyota    corolla      green      2007       7000
   Honda     civic       white      2006       6500

********** Class Dealer ******* Total cars = 9
  1. 1. Display all cars
  2. Display all cars within price range
  3. Display all cars with specific make
  4. Exit
>> 3
Enter the make you are looking for : Toyota
Make       Model       Color      Year       Price
```

```
 --------------------------------------------
  Toyota     Camry        red      2002      4000
  Toyota     Yaris        blue     2007      6500
  Toyota    corolla      green     2007      7000
********** Class Dealer ******* Total cars = 9
 1. Display all cars
 2. Display all cars within price range
 3. Display all cars with specific make
 4. Exit
>>
```

[5] **Restrictions and Suggestions**:
- Must define the two classes mentioned.
- The user input can be in either case.
- Since this is the first exercise that you are using strings, you are required to use both strings. For the auto and dealership classes, use string objects. In the getRecord(), string should be implemented as C-strings.
- Do not use global variables other than the one I listed and constants.
- Your main function should consist of primarily function calls. To make sure you are using functions properly.

[6] **Due Date**: Wednesday, April 15, 2015.