Assignment 2 – Arrays, Design, & Testing

You need to design, implement, and test a membership list program. The program should maintain and display a list of members.

You will use a class for the Members. **The class should have data elements for the following information:** Name, Date (of joining the club), Age, and Gender. The members should be listed and displayed in alphabetical order. How will you enter the name? What about hyphenated names? What about multiple names such a Ferdinand van der Smythe? Do you need individual get and set functions for each data member? Will you need a print function for each object? Any other functions?

You will also need a List class. **The List class will use a dynamic array to store Member objects.** As each item is entered an Item object must created and added to the array. What do you do if the array is full? One List object will have many Item objects. Do you need a print function in this class?

Your program must perform the following activities: create a list, add members, remove members, and display the membership list. To add an item you should prompt the user to enter the required information. **The display should show**: member name, date, age, and gender. Debug and test your program.

> ORIGINAL: Once you have the List and Item classes working correctly, test if an item is already in your List before adding it. **Overload the == operator to perform the test.** There is a simple example to overload this operator in the slides. Keep it simple. If that name is already in the list then simply display a message to the user. It IS their problem. ☺
>
> OPTION: Once you have the List and Item classes working correctly, test if an item is already in your List before adding it. Develop a member function to test if 2 names are equal. If that name is already in the list then simply display a message to the user. It IS their problem. ☺

You must create a design document. It should include the design of the classes and how you will use the classes. Remember to free memory if no longer used. Figure that out, **before** you start coding. If you do the design properly the coding should be easier.

You will provide a simple test plan. Since this is a program with input and output you should not need any driver functions. How do you handle spaces in names?

Part of testing is developing test data. Identify what data you will need for testing. Remember that boundary cases are the most important. Specify what data is needed for which tests.

NOTE: For testing please keep your array small, maybe only 4-5 elements. To test resizing you (and the grader) will need to insert only a few items into the list. It saves time for everyone! ☺

NOTE: Please use incremental development! Start with the List class and maybe a simplified Member class (only the item name?). If you're not comfortable with dynamic arrays start with a static array. Just remember to replace it later. Plan this out as part of your design. You will probably need to print the list early on so you can see what's going on inside the list and the items. When you have a (partial) program that compiles and runs properly save a copy!

You must **include a makefile and put all files for your assignment in a zip file**. If you do not do this assignment will NOT be graded.

Grading:

- programming style and documentation (10%)
- create the list class and object (15%)
- create the member class and objects (10%)
- add and remove members to the list (10%)
- maintain the list in alphabetical order (5%)
- you properly manage memory, i.e. NO leaks (5%)
- only notify the user when adding a duplicate member (5%)
- your array resizes properly (5%)
- Display the list with all data for each member on one line (10%)
- overload the equality operator (==) to prevent including duplicate items in the shopping list (10%)
- develop a design document (before you start coding), provide all the test data you used for testing, provide a test report (15%)