Program #1
CS 163 Data Structures

Submit your program to the D2L Dropbox

Scope: When beginning with this project, the first thing to keep in mind is that we have a short amount of time to complete each assignment. Therefore, it is critical that you focus on a limited scope, with an emphasis on the class(es) and data structures.

Our goal is to focus on data structures rather than on a comprehensive application program. *You will be primarily graded on your use of classes, member functions, arguments, data structures, pointers and the efficiency of your code.* Your program should compile and run and you should allow the grader to use any of your functions through a menu interface. Your user interface must be clear for us to thoroughly test all features. **It is not appropriate to hard code in the test cases - all tests should be interactive with the user.**

Background Information: With the stay home – stay safe requirements over the last few weeks, I have been ordering pretty much everything I need online. This ranges from groceries to computer gear. It has been very convenient, although I am concerned about not supporting the small stores that don't have an online presence. So, for this first programming assignment in CS163, we are going to develop software that will help small retailers be available to shoppers like myself who are shopping literally 100% from home.

Programming Assignment: This first program of the term is an exercise in building, traversing, and destroying linear linked lists. With program #1, you will build a linear linked list of stores. Each store will have a store name and website.

Then, for each store, there will be al linear linked list of the products that they have available to purchase. The items in this list should be grouped together (so all items of a particular category are together). Consider a struct for this information:

1. For each item keep track of the following information:
    a. The name of the item (e.g., Lightning cable)
    b. The category (e.g., Electronics)
    c. The price (e.g., 8.95)

Using Classes

We will be building a class (e.g., CS_Shop) to manage the data structures mentioned. You must have the following functions; the information that these functions need to work with should be passed as arguments. For example, to add an item for a store, the information about the item should be passed to the function. Example prototypes are supplied (you may change the prototypes)

a. Construct an object (constructor)
   CS_Shop ();
b. Add a store, passing in the store name and website
   int New_Store(char store_name[], char website[]);
c. Add an item in a store, pass in the store name and the information about the item being added. (store_item could be a struct)
   int Add_Item(char store_name[], store_item & to_add);
d. Remove an item, by name (passing in the name of the item to remove)
   int Remove_Item(char item_name[]);
e. Display all items offered at a store
   int Display_Store(char store_name[]);
f. Display all items that matcha particular category
   int Display_Category(char category[]);
g. Release all dynamic memory (destructor)
   ~CS_Shop ();

<u>Things you should know...as part of your program:</u>

1) You may use a combination of structures and classes.
2) Avoid long argument lists. Instead, package data into a struct and pass it to your member functions
3) Do not use statically allocated arrays in your classes or structures. All memory must be dynamically allocated and kept to a minimum!
4) All data members in a class must be private
5) All input operations should happen from the application program (called the "client)
6) Never output error messages from a class member function
7) Global variables are not allowed in CS163
8) **Do not use the String class! (use arrays of characters instead!);** however, you may use the cstring library of strlen, strcpy, strcmp
9) Use modular design, separating the .h files from the .cpp files. Remember, .h files should contain the class header and any necessary prototypes. The .cpp files should contain function definitions. You must have at least 1 .h file and 2 .cpp files.

Never implement functions in your .h file! And, never "#include" .cpp files!

10) Use the iostream library for all I/O; do not use stdio.h.

11) Make sure to define a constructor and destructor for your class. Your destructor must deallocate all dynamically allocated memory.

12) *Take a look at the style sheet which gives instruction on the topics that your write-up needs to cover.*