Program #1
CS 163 Data Structures

Submit your program to the D2L Dropbox

We do not accept late work beyond the late due date.

Scope: When beginning with this project, the first thing to keep in mind is that we only have approximately 2 weeks 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. *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; but we are not grading the "application". Instead your main program should be aimed at testing out your class member functions to make sure they are solid! Make sure to focus on thoroughly testing out your class member functions.

Therefore, focus on how to design classes that are well structured and efficient and on the required data structures. 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: The concept of having a smart home has really become popular. Think about it. From your phone you can now control your lights, turn on the coffee maker, preheat the oven, control the temperature in your home, run a vacuum cleaner, and much more. Over the break we installed a few of these items and it is fun to turn on or off everything without much effort. Of course, when writing an app to allow a user to control a smart home, care needs to be taken. In the app that I am using, although I can group lights together in a single room, each and every light can be managed independently which means my screen is cluttered with choices.

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 data structure for a smart home application to use. Although we are not building the application itself, we are working on data to allow an application to gain access to information it may need.

There will be two data structures required for this assignment:

1. A linear linked list of the different rooms that you want to control in a smart house. The names of these rooms can be set by the client program. For each room, we want to keep track of:
    a. The name of the room
    b. The list of accessories being controlled electronically

2. Then for each room, we want to have a linear linked list of all of the accessories (lights, coffee makers, heaters, security cameras, etc.) that can be controlled within that group. For each item keep track of:
    a. The name of the item (e.g. bedside lamp)
    b. The category of item (e.g., light)
    c. The status (e.g., on or off for a light, temperature for a thermostat, etc.)

The linear linked list rooms should be sorted by name.
The linear linked list of accessories should be grouped by category.

Using Classes
We will be building a class (e.g., CS_smart_home) 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 an argument. For example, to add an accessory, the information about the light switch needs to be sent as argument(s):

a. Construct an object (constructor)
b. Add a new Room, passing in the name of the room
c. Remove a Room (and all accessories associated with that room) given the name as an argument.
d. Add a new Accessory to a Room. Pass in the room name and information about the accessory being added.
e. Remove an Accessory from a Room by name
f. Display Accessories in a Room
g. Display all Accessories for All Rooms
h. Release all dynamic memory (destructor)

Things you should know...as part of your program:

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) Never prompt and read from the user when inside a class member function
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.*