Programming Assignment #3
CS 163 Data Structures

Submit your assignment to the D2L Dropbox (sign on via d2l.pdx.edu)

Programming – Goal: The goal of the third program is to create a hash table using chaining per Topic#6 and Lab #6. Hash tables are very useful in situations where an individual wants to quickly find their data by the "value" or "search key". You could think of them as similar to an array, except the client program uses a "key" instead of an "index" to get to the data. The key is then mapped through the hash function which turns it into an index!

Programming – Problem Statement: Have you ever wished you could quickly find an app on your phone, but it is nearly impossible. This is especially the case when you update an app and the widget used changes and is no longer recognizable. In fact, I would like to search for an app not by its name but what I can do with it. Not always easy to do. What I am not interested in doing is sorting the apps that I have available on my phone. So a hash table comes to mind as a good alternative.

We will be implementing a hash table using chaining as the collision resolution technique that uses a list of applications that are on a mobile device. Your program will hash on the keyword used in the description to determine the match. Each item in our table will have the following information (at a minimum) which should be stored as a struct or a class:

1) Application name
2) List of at most 5 keywords that describe why this app is used (e.g., business, books, education, mapping, etc.)
3) A description of what can be done with this application
4) A rating of how you have enjoyed using the app in the past

Data Structures: Write a C++ program that implements and uses a **table abstract data type using a hash table (with chaining)** to add applications, retrieve (e.g., find a match based on a keyword), and display (all matches and display all). Since there will be multiple instances of an app in the table (because of multiple keywords), your nodes should have pointers to the app so that only one instance of the memory exists.

What does retrieve need to do? It should take one of the keywords and supply back to the calling routine all the apps that have that keyword in their list. Retrieve, since it is an ADT operation, should not correspond with the user (i.e., it should

not prompt, echo, input, or output data). Retrieve should have an array of apps as an argument that is filled by the ADT. The apps may be stored as a struct or class.

Evaluate the performance of storing and retrieving items from this table. Monitor the number of collisions that occur for a given set of data that you select. Make sure your hash table's size is a prime number. Try different table sizes, and evaluate the performance (i.e., the length of the chains!). **Your design writeup must discuss what you have discovered.**

In summary, the required functions for your TABLE ADT are:

1. **Constructor,**
2. **Destructor**
3. **Insert a new app**
4. **Retrieve** *(based on a keyword)*
5. **Display** *(only displaying matches, based on a keyword)*
6. **Display all**

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

1) Do not use statically allocated arrays in your classes or structures. All memory must be dynamically allocated and kept to a minimum!
2) All data members in a class must be private
3) Never perform input operations from your class in CS163
4) Global variables are not allowed in CS163
5) Do not use the String class! (use arrays of characters instead and the cstring library!)
6) 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 "#include" .cpp files!**
7) Use the iostream library for all I/O; do not use stdio.h.
8) Make sure to define a constructor and destructor for your class. Your destructor must deallocate all dynamically allocated memory.
9) Remember that 20% of each program's grade is based on a written discussion of the design. *Take a look at the style sheet which gives instruction on the topics that your write-up needs to cover.*