Programming Assignment #4
CS 163 Data Structures

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

Programming – Goal: The goal of this program is to create a binary search trees (BST) and to implement the insert, retrieve, display, remove all matches by keyword, and remove all. All of these functions should be implemented **recursively.**

Background: Take your program #3 for finding apps and replace the use of a hash table now with a binary search tree, organized by the keyword. Retrieve and remove by keyword used in the description, should retrieve and remove any matches for that keyword as well. Of course there should be a constructor and destructor.

Data Structures: Create an **abstract data type using a Binary Search Tree**. This time we will also get our data sorted when we traverse using inorder traversal!

What does retrieve need to do? It needs to supply back to the calling routine information about the item that matches. Retrieve, since it is an ADT operation, should not correspond with the user (i.e., it should not prompt, echo, input, or output data).

In your design writeup, discuss the difference between solving the problem with hash tables (program #3) versus binary search trees (program #4).

Each item in our tree 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

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. **Remove** *(based on a keyword)*
6. **Display** *(only displaying matches, based on a keyword)*
7. **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.*