Lab 8: C++
Classes with Dynamically Allocated Data Members

Turn in:

1) WordList.h
2) WordList.cpp
3) FlashCards.cpp
4) makefile

Highly recommended:

Download, read, compile, and run the example programs pointer4.cpp and stringpt.cpp before you start this assignment.

Warning: Working together on assignments is considered cheating in this class. Copying code from books, web sites or other sources is also considered cheating.

Cheating will result in a grade of zero for the assignment.

WordList
- wordCount : int - wordPtr : string * [ ] an array of pointers to string objects
<<constructor>> WordList(count : int) ~WordList( ) + getWordCount( ) : int + insertWord (sp : string* , index : int) + shuffle ( ) + sortByLength( ) + sortByAlpha( ) + print( )

Assignment:

WordList.h and WordList.cpp: Create C++ programs to implement the class shown in the UML diagram above. Additional information on member functions:

1) There is only one constructor. The constructor should set wordCount to the parameter value and then dynamically allocate an array of string pointers using wordCount as the number of elements. After the array is allocated, the constructor should set each element of the array to NULL.
2) There is no default constructor.
3) ~WordList is the destructor. The destructor should delete the dynamically allocated memory.
4) getWordCount is a standard accessor. (Note: There is no mutator for wordCount.)
5) insertWord – This function should store a given pointer (1st parameter) in the array at the given subscript (2nd parameter). The function MUST do bounds checking. If the 2nd parameter is out of bounds, do not change the array. [[You may print an error message for debugging purposes.]]

6) shuffle – Make 20 random swaps.
7) sortByLength – put the array elements in order by the lengths of the strings to which they point. You can use any sort algorithm you want. Relational operators < > <= >= are overloaded for the string class.
8) sortByAlpha – put the array elements in order by the alphabetical order of the strings to which they point
9) print – outputs the strings, one per line

FlashCards.cpp: Create a test program. In the main function, do the following:

1. Ask the user for the number of words.
2. Create a WordList object with that number of words.
3. Ask the user to input the words and insert them into the WordList object.
4. Print the list of words.
5. shuffle the words then print the list again.
6. Sort the words by length then print the list again.
7. Sort the words by alpha then print the list again.

More information:

C++ string class Tutorial at http://www.cplusplus.com/reference/string/string/

#include <string>

Dynamic memory allocation Reference: http://www.cs.fsu.edu/~jestes/cop3330/notes/dma.html