Assignment P4

Due Date: March 5

Purpose

In this project, you will bring together all of the basic elements of C++ (sequential, selection, and repetition statements, as well as functions). In addition, this will be your first object-oriented program that includes a class and member functions. Arrays will be used to store data needed to solve the problem.

Problem

Miguel Trout, an English professor of some renown at a small liberal arts college has a strange way of grading. He grades papers according to a formula that includes the number of times a word is used. Professor Trout enlists you, an expert in C++, to write a program that will keep track of the number of times words are used in an arbitrary text that the user specifies.

Input

The program should prompt for the name of a file that contains the text to process. The file name can be any length, but is assumed to not include any spaces. The text file will contain one normal English paragraph, including punctuation. The word "DONE" (in all capital letters) denotes the end of the file, but should be ignored in the remaining processing.

Output

The program should display a list of all of the words found in the text along with the number of occurrences of that word. Words that are used only once should be listed separately after all those that appear multiple times. For full credit, the list should be ordered according to the number of occurrences. Note that this does **not** mean the words have to be sorted (see below). Capitalization and punctuation should be ignored; i.e., "Hello" is the same word as "hello." For example, if the input was:

```
Computer Science is swell.  The computer is my pal.
Is it yours? DONE
```

The output would be something like:

```
Word          Number of Occurrences
==================================
is                    3
computer              2

Words with one occurrence
==================================
science swell the my pal it yours
```

Specifics

- Your `main()` function **must** be *exactly* as follows (comments/spacing notwithstanding):

```
int main () {
   trout myList;
   string fileName;

   cout << "Enter file name: ";
   cin >> fileName;
   myList.readText (fileName);
   myList.displayStats();
   return 0;
}
```

- Following the code above, you must have a class called `trout` that contains all of your data members and methods. You must have two public methods called `readText()`, which reads in the data and does most of the processing, and `displayStats()`, which displays the results.
- You may have additional private member functions if you wish.
- The program should test if the file name given by the user is valid. An error message should be displayed if not, and the program should stop.
- The data file will always contain the word "DONE." Valid punctuation includes periods, commas, exclamation points, and question marks. You may assume that apostrophes are part of a word.
- The output should be aligned as in the example shown above. Words with more than one occurrence should be left justified. The number of occurrences should be right justified. Words with only one occurrence should be displayed with one space in between each word, **with at most 10 words on a line.**
- There is some string processing to do here. Some functions in particular that may come in handy are `length()`, `erase()`, and `c_str()`.

Notes

- Write your program in stages. The algorithms necessary are not particularly difficult, but there are many small details to worry about. Start by reading the data and simply displaying all of the words, punctuation and all. Then add in code, step-by-step, to perform the various necessary tasks. In this way, you should always have a working program, or one that works up to the newest item you just added. Debugging is much easier using this method.
- As usual, the name of your **source code** file should be your last name and project number; e.g., gousieP4.cpp.
- As usual, email your source code to mgousie@wheatoncollege.edu (or gousie_mike@wheatoncollege.edu). Submit by 11:59:59 PM on the due date for your project to be on time.
- Hand in a printout of your source code in class the next day. Write/print and **sign** the Wheaton Honor Code Pledge on what you turn in: "I have abided by the Wheaton College Honor Code in this work."

...prose = words in their best order;
poetry = the best words in their best order.
– Samuel Taylor Coleridge (1772-1834)