**Instructions:**

Download the file hw3a-files.zip.

**Objectives**

The purpose of this assignment is to get some familiarity with the symbol table abstract data type by using one to build a class that can be used to find the frequency counts of words appearing in text files. The class is called WordFrequencyAnalyzer and this is the primary class you will be developing.

**Files**

* The directory datafiles contains the text files used for your tests. The files in the directory need to be placed at a suitable place in your directory structure so that your program is able to read it.
* **HW3aTest** – This is a JUnit test for your solution. Each test method has a number in its name. The sum of all the numbers in all the tests that pass will give you your base score on the assignment. Note that this unit test is used for grading. They may not be the best tests to use for debugging. You will probably want to write your own tests when debugging.
* **WordFrequencyAnalyzer** - This is the primary class you will be implementing. You need to implement the class by providing definitions for the constructor and methods. Note that the purpose of the field called **counters** is to keep track of (store) how many times each word appears in the given text file.
* **WordFrequencyAnalyzerClient** – This class uses the WordFrequencyAnalyzer class you are implementing to find the frequency of a few specific words in a couple of text files as well as to determine the number of times the most frequently occurring word appears in each file. You are not asked to modify or use this class in any way. It is provided solely to demonstrate how the WordFrequencyAnalyzer class you are implementing could be used.

**Restrictions**

* The WordFrequencyAnalyzer.java file contains a single field of type SequentialSearchST<Word, Integer> called **counters** that you will use to store the counts of all the words appearing in the text file. You are not allowed to change this declaration in any way nor are you allowed to add any other fields to the WordFrequencyAnalyzer class.
* you must count the frequencies of all the words in a single pass through the file done at the moment that a WordFrequencyAnalyzer object is created and you are not allowed to access the text files in any way when getCount or maxCount are called. The counts of all the words should have already been calculated and stored in the counters field when the WordFrequencyAnalyzer object was constructed.
* Use of the readString method from <https://introcs.cs.princeton.edu/java/stdlib/javadoc/StdIn.html> might be useful..

**WordFrequencyAnalyzer API**

* **WordFrequencyAnalyzer(String filename)** - The constructor takes the name of a file and then processes the text file, keeping track of all the words appearing in the file and their counts.
* **int getCount()** - The getCount method is only invoked after the file has been processed (after the constructor has finished executing). If the constructor correctly populated the symbol table, you can just get the counts directly out of the symbol table. You are not allowed to read from the file when getCount is called.
* **int maxCount()** - The maxCount method will require slightly more work. You will need to loop over the contents of the symbol table to determine the count for the most frequently appearing word. You are not allowed to read from the file when maxCount is called.

**Submission**

Submit the **WordFrequencyAnalyzer.java** file. These are the only two source files you are allowed to submit, so don’t add any other files nor modify any other files.