CISC 1115
Brooklyn College
Professor Langsam

Assignment #6

Write a Java program that encodes English-language phrases into pig-Latin. Pig-Latin is a form of coded language often used for amusement. According to Wikipedia:

> **Pig-Latin** is a language game primarily used in English. An alternative British name for Pig-Latin is **backslang** (in Britain this term more often applies to the type of backslang used by the criminals of 19th century London and used as a playground game today, which was based on turning words backwards), or Butcher's Backslang which was common in English Butcher's shops at least until World War II. Prior to this, Benjamin Franklin was known to use a version of Pig-Latin in some publications. Pig-Latin is usually used by children for amusement or to converse in (perceived) privacy from adults or other children. Conversely, adults sometimes use it to discuss sensitive topics they do not want very young children to overhear.

Many variations exist in the methods used to form pig-Latin phrases. For simplicity, use the following algorithm:

- To form a Pig-Latin phrase from an English-language phrase, tokenize the phrase into words. Place the first letter of the English word at the end of the English word and add the letters "ay." Thus the word "jump" becomes "umpjay," the word "the" becomes "hetay" and the word "computer" becomes "omputercay." Blanks and punctuation marks between words remain unchanged. If a word in the English phrase was capitalized you will have to change the case of the Pig-Latin phrase accordingly. Thus the word "Professor" would become "Rofessorpay."

Use the text file "GettyburgAddress.txt" found at

http://eilat.sci.brooklyn.cuny.edu/cisc1115/CISClassPage.htm

At the end of your program print the following statistics:

1. The number of words in the original text.
2. The number of letter in the original text.
3. The number of punctuation marks in the original text.
4. The number of whitespace characters in the original text.
5. The number of all characters in the original (including punctuation, digits, and whitespace.)

Strategy:

1. Write a method, *convertWord,* which given a word in English returns the equivalent word in Pig-Latin.
2. Write a method, *processWords,* which reads a word one at a time from the input file and sends it to the *convertWord* method. After returning from the *convertWord* method the *processWords* method should print the word and after all words have been processed, print the number of words processed by the method. Print 10-words per line. It should then close the input file.

3. Write a method, *countItemsInFile*, which reopens the original input file. This method should read an entire line at a time from the file, processing each character in the string to determine if it is a letter, digit, punctuation mark, or whitespace. It then prints the results.
4. The main method invokes the *processWords* method and then invokes the *countItemsInFile* method.

Suggestions:

- While developing and testing your program use a smaller datafile than the one specified above.
- You may modify the character and line processing routines we developed in class.

Be sure to use the structured programming techniques we have learned in class. Do not use global variables. Use meaningful variables and comment each method.