CITC-1301 Introduction to Programming

Chapter 9 Lab – Text File Analyzer

Write a Python program that asks the user for a file name, reads the text from the file, and calculates the approximate number of sentences, unique words, and total words contained in the text. Additionally, calculate and output the frequency of each word sorted from the most occurrences to the least occurrences.

Design Requirements

- Create a **`main()`** function containing the mainline logic of the program.
- In the **`main()`** function, validate that the file specified by the user exists before reading the file. If the file does not exist, inform the user and ask for a file name again until the user provides good input.
- Use a **`try...except`** statement to open the specified file, read from the file, and close the file.
- Use the **`sortDictionary()`** function below to sort the dictionary object containing the word statistics.

Function for sorting a dictionary object in reverse order:

```
# ------------------------------------------------------------------------------------------
#                                  sortDictionary(dictionary)
# Function sorts passed in dictionary object based on keys/values in reverse alphabetical order
# and returns sorted list of tuples.
#   Usage:
#       testDictionary = {}
#       testDictionary["One"] = 5
#       testDictionary["Two"] = 15
#       sortedList = sortDictionary(testDictionary)
#       print(sortedList)
#       Output: [('Two', 15), ('One', 5)]
# ------------------------------------------------------------------------------------------
def sortDictionary(dictionaryObject):
    return sorted(dictionaryObject.items(), reverse=True, key=lambda kv:kv[1])
# end method
```

Design Suggestions

- Read entire text file into a variable. Set the text to lowercase.
- Count the number of periods, question marks, and exclamation points to calculate the approx. number of sentences by using the string object's **`count()`** method or by looping through the variable containing the file text one character at a time.
- Remove *all* punctuation from the text (e.g., commas, apostrophes, periods, question marks, exclamation points, single- and double-quotations, etc.) by using the string object's **`replace()`** method.
- Split the text into a list by using the string object's **`split()`** method. This list now contains each word from the text (with duplicates).
- Using the list object with duplicates created above, create another list object to hold only one instance of each word by passing the list object with duplicates to the **`set()`** function. Additionally, pass the result of the **`set()`** function to the **`sorted()`** function to sort the list object in alphabetic order.
- Loop through the sorted list object with no duplicates created above to add each unique word to a dictionary object (as the dictionary keys) and setting the counts for each word to zero (as the dictionary values).

- Loop through each word in the list object with duplicates to count the number of instances of each word by incrementing the values of each word's entry in the dictionary object created in the previous step.
- Us the **`sortDictionary()`** function to sort the items in the dictionary object.
- Output the number approximate number of sentences, the number of unique words, the total number of words, and the sorted dictionary listing each word and the number of instances of each word.

Example text file:

```
Three Rings for the Elven-kings under the sky,
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie.
One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie.
```

Example output:

```
Text File Analyzer

File path: rings.txt [ENTER]

Number of sentences: 2
Number of unique words: 40
Total number of words: 74

Word               Count
----               -----
The                9
For                4
In                 4
One                4
Them               4
...
```

Submission Instructions

- Upload your Python script (i.e., your .py file) to the appropriate dropbox on D2L.