DSC510 Assignment 8

Directions

We will create a program which performs three essential operations. It will process a txt file (Gettysburg.txt).

Open the file and process each line.

Either add each word to the dictionary with a frequency of 1 or update the word’s count by 1.

Nicely print the output, in this case from high to low frequency. You should use string formatting for this. (See discussion 8.3).

We want to achieve each major goal with a function (one function, one action). We can find four functions that need to be created.

add_word: Add each word to the dictionary. Parameters are the word and a dictionary. No return value.

Process_line: There is some work to be done to process the line: strip off various characters, split out the words, and so on. Parameters are a line and the dictionary. It calls add word with each processed word. No return value.

Pretty_print: Because formatted printing can be messy and often particular to each situation (meaning that we might need to modify it later), we separated out the printing function. The parameter is a dictionary. No return value.

main: We will use a main function as the main program. As usual, it will open the file and call process_line on each line. When finished, it will call pretty_print to print the dictionary.

In [1]:
```
# File:   DSC510_Assignment_8.py
# Name:   Kevin Paulovici
# Date:   1/27/19
# Course: DSC 510 - Introduction to Programming
# School: Bellevue University
# Desc:   This module is for week 8 programming assignment.

import string
```

In [2]:
```
# Function: main
#
# Desc: Function opens a file and passes each line and
#       a dictionary to Process_line.
#       When the file is done reading it will send the
#       dictionary to be printed.
#
# Pre:  The function requires a valid file.
#
# Post: None
def main():
    # define dictionary of words and frq
    word_dict = {}

    with open('Gettysburg.txt') as file_in:
        for line in file_in:
            Process_line(line, word_dict)

    # print out the dict
    Pretty_print(word_dict)
```

In [3]:
```
# Function: Add_word
#
# Desc: Function takes a word and dictionary from Process_line.
#       The word is checked if it is in the dictionary. If yes,
#       the count is incremented. If no, the value is added and
#       incremented.
#
# Pre:  The function requires a line from a file.
#
# Post: None
def Add_word(word, word_dict):
    # add words and update freq. to dictionary
    word_dict[word] = word_dict.get(word, 0) + 1
```

In [4]:
```
# Function: Add_word
#
# Desc: Function takes a word and dictionary from Process_line.
#       The word is checked if it is in the dictionary. If yes,
#       the count is incremented. If no, the value is added and
#       incremented.
#
# Pre:  The function requires a line from a file.
#
# Post: None
def Add_word(word, word_dict):
    # add words to dictionary
    word_dict[word] = word_dict.get(word, 0) + 1
```

In [5]:
```
# Function: Process_line
#
# Desc: Function takes a line read from a file and a dictionary
#       from main. The line is cleaned to remove punctuation and
#       set to lower case. Each word in the line is sent to the
#       Add_word function.
#
# Pre:  The function requires a line from a file.
#
# Post: None
def Process_line(line, word_dict):
    # clean up the line
    line = line.translate(line.maketrans('', '', string.punctuation))
    line = line.lower()
    # pass word and dict to be added
    for word in line.split():
        Add_word(word, word_dict)
```

In [6]:
```
# Function: Pretty_print
#
# Desc: Functions prints out a dictionary.
#       The dictionary is converted to a list of tuples
#       to be sorted by frequency of use (high to low)
#
# Pre:  The function requires the dictionary to be set.
#
# Post: None
def Pretty_print(word_dict):
    # add dict to list of tuples and sort
    word_list = []
    for key, val in word_dict.items():
        word_list.append((val, key))
    word_list.sort(reverse=True)

    print("Length of dictionary: {}".format(len(word_dict)))
    print('{:15} {}'.format('Word', 'Count'))
    print('-' * 22)

    for item in word_list:
        print('{0[1]:17} {0[0]}'.format(item))
```

In [7]:
```
main()
```

```
Length of dictionary: 143
Word                Count
----------------------
that                  13
the                   11
we                    10
to                    8
here                  8
a                     7
and                   6
of                    5
not                   5
nation                5
it                    5
have                  5
for                   5
can                   5
this                  4
in                    4
```