CS 153 / 453
Lab Assignment # 6

All code you submit must be your own individual work.

Header comments and inline comments are required.

Tutorial

- To open a file for reading:

```
myFile = open("file1.txt","r")
```

- To read a line from the file:

```
line = myFile.readline()
```

Note: The string will contain a newline character as the last character.

- To check whether the readline was successful:

```
if not line:

   # there was no more input in the file, readline failed
```

- To close the file:

```
myFile.close()
```

- A file object is iterable. It can be used in a for loop with the in membership operator.

Here's an example of how these steps can be used to read all of the lines from a file.

```
filename = "file1.txt"
myFile = open(filename, "r")
for line in myFile:
   print(line)

myFile.close()
```

Tutorial Continued

So far, we haven't defined a "main" function. The reason is because the programs we've written have been stand-alone programs.

Now, we need to consider the situation where our programs will be part of a larger system. Starting with this assignment, you'll define a function called main that contains the code for the main program.

In the main program area, you'll have a statement that determines whether this program is running standalone (or is being used as a module in another program).

First, define the main function.

```
def main( ) :
     # place all statements for the main program here
```

Then, place an if statement in the file to determine if the program is running standalone.

```
if __name__ == '__main__':

     # call the main function
     main( )
```

Note: there are two underscores before and after name and before and after main.

Other Information You Will Need

Some Python string methods:

isalnum()	Returns True if all characters in the string are alphanumeric
isalpha()	Returns True if all characters in the string are in the alphabet
isdigit()	Returns True if all characters in the string are digits
isspace()	Returns True if all characters in the string are whitespaces
isprintable()	Returns True if all characters in the string are printable

There is no Python method called ispunct in the core Python language.   You may find a module online that contains such a method but you are not permitted to use it in this assignment.

The Assignment

Part A:

Create three text files to use for testing your Python program. In each file, type several sentences. Include punctuation symbols, spaces, upper- and lowercase letters.

Part B:

Create a Python program named lab6.py.

Use incremental programming. Write one function at a time and test that one function.

When the program is finished, you will have 4 functions. Test your program thoroughly using at least 3 text files (created in part A).

Function 1: remove_punc(s)

A punctuation symbol is any printable character that is not a digit, not whitespace, and not a letter. (see the section above on string methods)

- parameter: a string
- return value: a new string that contains the original string with all punctuation symbols removed.

Function 2: word_count(s)

- parameter: a string that contains one or more words, separated by spaces (there are no punctuation symbols in the string)
- return value: an integer, the number of words in the string

The function should count and return the number of words in the string.

Function 3: add_unique(s, word_list)

- parameters: 1) a string that contains one or more words, separated by spaces (there are no punctuation symbols in the string)
  2) a list of strings
- return value: none

The function should split the string into words. Then, the function should append the lowercase version of the words to word_list **if they aren't already in the list**.

Function 4: the main function

Pseudocode:

```
 1.  Input the name of a file from the keyboard
 2.  Open the file for reading
 3.  Initialize a count for number of punctuation symbols in the file
 4.  Initialize a count for the total number of words in the file
 5.  Initialize an empty list to hold the list of unique words in the file
 6.  Repeat for each line in the file
       a) Determine the number of punctuation symbols in the line and add to count
       b) If there's a newline at the end of the line, discard it.  Then determine the number of
          words in the line and add to count.
       c) Add any new words from this line to the list of unique words.
 7.  Display the number of punctuation symbols in the file.
 8.  Display the total number of words in the file.
 9.  Display the number of unique words in the file.
10.  Sort the list of unique words.
11.  Display the sorted list of unique words in 3 columns.  (see example input and output
     below)
```

After the main function, include the if statement to check whether the program is being run standalone. (See the Tutorial above.)

Do not write any other functions.

Test thoroughly. Debug if needed.   When it works correctly, check to make sure that you have documented the program with header comments and inline comments.

Submit lab6.py on Canvas.

Example Input File:

```
Depending on who you ask, there are nine (or eight) planets in the
solar system. We're on the 3rd planet from the sun.
```

Expected Output:

```
Number of Punctuation Symbols: 6
Total Words:  23
Unique Words:  20

Unique Words

3rd                        depending                  in
are                        eight                      nine
ask                        from                       on
```

or
planet
planets
solar
sun
system
the
there
were
who
you