

CS 153 and CS 453 Lab Assignment # 7

Refresher: Getting Input From a File

- To open a file for reading:

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

- To read one 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()
```

Usually, we want to read all of the lines from a file and do the same thing with each line. A file object is iterable. It can be used in a for loop with the in membership operator.

Here's an example of how a loop can be used to read all of the lines from a file (one at a time).

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

myFile.close()
```

Writing Text Output to a File

- To open a file for writing:

```
outputFile = open("file2.txt", "w+")
```

The w is for "write" and the + means it will create the file if it doesn't already exist.

- To write text data to the file, you can use the **write** function:

```
outputFile.write( someString )
```

The parameter to the write method must be a string. If you want a newline in the file, you have to include a newline in the string you write.

Save time. Use incremental programming. Write one function. Test that function. Repeat until done.

- When you're finished writing text to the file, be sure to close the file:

```
myFile.close()
```

Appending Output to a File That Already Exists

- To open a file for append, the only thing that's different is the file mode. Use **a** for append instead of **w** for write.

```
outputFile = open("file2.txt", "a+")
```

The Input Files

Using a text editor such as Notepad, TextEdit, or Kate, open the file file1.txt and familiarize yourself with the contents. Close the file.

Using a text editor such as Notepad, TextEdit, or Kate, open the file file2.txt and familiarize yourself with the contents. Close the file.

Do not change these files.

The Assignment: Name your program lab7.py

- **Write the 6 functions below. Name the functions exactly as shown.**

```
def remove_non_alpha( s ) :
```

```
    # s is assumed to be a string.
    # Return a string that contains only the alphabetic
    # characters (letters) of s.
```

```
def remove_non_numeric( s ) :
```

```
    # s is assumed to be a string.
    # Return a string that contains only the characters of s
    # that can form a valid int or float number. Remember that a
    # numeric value can have a leading '+' or '-' and a float can
    # have a decimal point (but only one decimal point).
```

Need to know more about List Comprehensions?

Here's a pretty [informative web page](#).

```
def list_only_numbers( a_list ) :

    # a_list is assumed to be a list of strings.
    # Create a new empty list.
    # Using a loop (or a list comprehension)
    # 1) call remove_non_numeric with a list element
    #    a) if the return value is not the empty string, convert
    #       the string to either int or float (if it contains a decimal
    #       point) and append the value to the new list.
    #    b) if the return value is the empty string, do not
    #       append anything to the new list.
    # Return the new list.

def only_letters ( a_list ) :

    # a_list is assumed to be a list of strings.
    # You must use a list comprehension and call your remove_non_alpha
    # function.
    # Create and return a new list that contains strings that have
    # only the letters (non-alphabetic characters have been removed)
    # from the elements of a_list.

def list_stats( a_list ) :

    Follow the pseudocode steps below.
    1) Find the sum of the elements in a_list and store it in a variable
       named list_sum.
    2) Find the mean (average) of the elements and store it in a
       variable named mean.
    3) Use a list comprehension to create a new list that applies the
       expression (i - mean) ** 2 to each element i of a_list. Store the
       new list in a variable named squares.
    4) Find the sum of the elements in squares and store it in a variable
       named sq_sum.
    5) Calculate the standard deviation.

       std_dev = square root of (sq_sum / (N - 1))

       where N is the number of elements in the original list

    6) Return a list that contains [ list_sum, mean, std_dev ]
```

```
def main( ) :
```

- 1) Open file1.txt for reading. This is the input file.
- 2) Open output1.txt for writing. This is the output file.
- 3) Reminder: each line from the input file is a string.
Use a loop that reads lines from the input file and for each line, do the following:
 - a) Use a list comprehension - create list3A, split the string and convert each element to lowercase.
 - b) Use a list comprehension - create list3B, a list that has all non-alpha characters removed from the elements of the list obtained in step 3a. (Do not combine steps 3a and 3b.)
 - c) Use a list comprehension - create list3C, a list that contains the length of each element in the list obtained in from step 3b. (Do not combine steps.)
 - d) Find the average length of the elements in list3B. (Do not combine steps.)
 - e) Write output to the output file (black font indicates text that should appear, blue font indicates program data to be printed):

Original line: original line from the input file

Words: elements of list3A with one space between elements

Only Letters: elements of list3B with one space between elements

Average word length: average

---print a blank line---

- 4) When the loop is finished, close both files.
- 5) Open file2.txt for reading.
- 6) Open output2.txt for writing.
- 7) Use a loop that reads lines from the input file and for each line, do the following:
 - a) Create list7A by splitting the string (no parameter on the split method means that it splits using whitespace as the delimiter).
 - b) Create list7B by calling the list_only_numbers function using list7A.
 - c) Call the list_stats function, pass it list7B, and store the return value in a variable called **stats**.
 - d) Write output to the output file (black font indicates text that should appear, blue font indicates data to be printed):

Original line: original line from the input file

Numeric Data: elements of list7B with one space between elements

Sum: sum of the elements of list7B

Mean: mean of the elements of list7B

Std Dev: standard deviation of the elements of list7B

---a blank line---
 - e) When the loop is finished, close both files.

- After the main function, include the if statement that checks to see if this program is being run stand-alone.
- Include header comments as in previous lab assignments. Include inline comments in each function. (Hint: the pseudocode above makes good inline comments)

Submit lab7.py on Canvas.