**Assignment Directions**

For all assignments, follow the guidelines in the PEP8 Standards for this Class document in the Student Docs section. Your programs and zip files must be named correctly, contain a docstring heading and include the other style requirements taught to this point. Be sure that your user prompts and print outputs are descriptive.

The specific instructions for these problems are adapted from problems shown in the textbook. Make sure to follow the specific instructions in THIS document.

**Chapter 7 Exercises**

**4.7.1:** Given a constant list of integers in the range 1 to 10 inclusive, **use list comprehension** (no explicit loops) to:

* find the sum of the even integers in list L.
* find the sum of the odd integers in list L.

Example of Output:  
Evaluating the numbers in: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
Even: 30  
Odd: 25

**4.7.2:** Given a constant list of integers, write Python code to generate a new list with same number of elements as the original list such that each integer in the new list is the sum of its nearest neighbors and itself from the original list. Print both lists with descriptions.

Example of Output:  
Input List: [10, 20, 30, 40, 50]  
Result List: [30, 60, 90, 120, 90]

**Chapter 9 Exercises**

**4.9.3** Start with 2 constant lists. One with first names and another of last names.   
  
First validate that both lists are the same size and if not, exit with an error message.   
  
Use zip to create a dictionary with the keys as the last names and the values as the first names. Print the generated dictionary with an appropriate description.

Example of Output:  
First Names: ['Jane', 'John', 'Jack']  
Last Names: ['Doe', 'Deer', 'Black']  
Name Dictionary: {'Doe': 'Jane', 'Deer': 'John', 'Black': 'Jack'}

**4.9.4:** Using my\_dict = {a':15, 'c':18, 'b':20}, write a program to:

1. print all the keys.
2. print all the values.
3. print all the keys and values pairs.
4. print all the keys and values pairs in order of key.
5. print all the keys and values pairs in order of value.

*\* Ordering may be ascending or descending*

Example of Output (showing alternative ways **you might choose** to print lists and dictionaries):  
a. Keys: ['a', 'c', 'b']  
b. Values: 15, 18, 20  
c. Key value pairs: a: 15, c: 18, b: 20  
d. Key value pairs ordered by key: [('a', 15), ('b', 20), ('c', 18)]  
e. Key value pairs ordered by value: a: 15, c: 18, b: 20

**4.9.5:** Write the following python program.   
  
Assign to a constant variable an English sentence of at least 15 characters.  
  
Print the starting sentence

Create a **dictionary** withthe letters as keys and frequency counts as values.  
Print the dictionary.  
  
Create a **string** of the most common letter(s) in the sentence. In the case of a tie for the most common letter, the string will have all of them  
Print the string and the number of times the letter(s) appeared in the sentence  
  
Create a list of the unique letters, with each letter being the repeated number of times it appears in the string. Then print this list as one element per line (a histogram).

Example Output #1  
The string being analyzed is: "WAS IT A RAT I SAW"  
1. Sorted dictionary of letter counts: {'A': 4, 'I': 2, 'R': 1, 'S': 2, 'T': 2, 'W': 2}  
2. Most frequent letter "A" appears 4 times.  
3. Histogram:  
 AAAA  
 II  
 R  
 SS  
 TT  
 WW

Example Output #2  
The string being analyzed is: "WWAS IT A RAT I SAWW"  
1. Sorted dictionary of letter counts: {'A': 4, 'I': 2, 'R': 1, 'S': 2, 'T': 2, 'W': 4}  
2. Most frequent letters ['A', 'W'] appear 4 times.  
3. Histogram:  
 AAAA  
 II  
 R  
 SS  
 TT  
 WWWW

**4.9.6:** Create a program that:

* prompts a user for a number
* validates the number
* re-prompts on error
* converts the number to words using a dictionary
* prints out the converted numbers as words

The program must only have one input command and work for any size positive or negative number.

Decimal point should be converted to ‘point’.

If the user enters commas, tell them to try again without the commas.

Example Output #1  
Enter a number: 123  
As Text: One Two Three

Example Output #2  
Enter a number: -123  
As Text: Negative One Two Three

Example Output #3  
Enter a number: 1234.76  
As Text: One Two Three Four Point Seven Six

Example Output #4 – invalid Input  
Enter a number: 1,000  
Please try again without entering commas.  
Enter a number: 1 thousand  
"1 thousand" is not a valid number. Please try again  
Enter a number: 1000.00  
As Text: One Zero Zero Zero Point Zero Zero

**Where to submit?**

Click Assignments in the Navigation Area and then click on the title of the assignment to enter the submission area and upload the zip file with your response.