CS-521 Homework Assignment 5

Assignment Directions

Complete the 5 python coding problems below. Each is worth 10 points for a total of 50 points (100%).

Individual programs must be named with your BU email prefix (the part before @bu.edu) and the problem number. If your email is alex@bu.edu, then your first program in this assignment would be called: alex_hw_5_1.py

The programs must all be combined into a single zip file named with your email prefix and the assignment number. alex@bu.edu would name their submission alex_5.zip.

Style Requirements

For all assignments, follow the guidelines in the PEP8 Standards and Best Practices that have been shared to date, along with course specific requirements. Remember these to avoid minor deductions. In **bold** are new style requirements since the previous module.:

- Name programs and zip container correctly
    - Among other things, names for regular programs and zip file must be all lower case.
- Include a program docstring
- Stay under 80 characters on all code and comment lines
- Ask for input() with descriptive prompts telling users what is expected
    - Especially remember to tell user what delimiter you're expecting when using split()
- Make sure to validate user input for data type, value range and other problem requirements
- Print output that clearly explains what is being printed (where necessary)
    - In other words, don't just print a '5' unless it's clear what that 5 represents.
- Import statements should be immediately after the program docstring
- Make sure to include # line comments (just a few in each program, don't go crazy)
- Use snake_case (lower case plus underscores) for variables
- CONSTANT variables must be all upper case and immediately following any import statements
- **Provide a docstrings for your user defined functions objects**
- **Trap errors from bad user input that might cause code to crash**
- **Keep scope for try blocks VERY small – generally one or two commands**
- **Use format() or format strings f' ()' to round output and add thousand separating commas (US standards)**
    - 123456.126 should be 123,456.13 ➔ '{:,.2f}'.format(123456.126)
- **NEVER use a java/c++ style "main()" function**
    - Or any function that neither has input arguments nor return values

You are now expected to *create user defined functions when called for in this assignment.*

CS-521 Homework Assignment 5

Modules Allowed
Unless noted in a question, you are welcome to import the following modules into your programs and use their methods and attributes without special permission from your facilitator.

Do not import a module unless you are using it.

- operator
- os
- random
- re
- string
- sys

CS-521 Homework Assignment 5

Programming Problems

The following includes concepts taught in Chapter 5

5.1: You are going to ask a user for a text file and then count the number of vowels and consonants in that file. Write a python program that does the following:

- Continually prompts user for a text file to read until a valid file is supplied
- Create a function named vc_counter():
  - takes the valid file name as an argument
  - counts the number of vowels and consonants in the file
  - returns a dictionary of the count values, using the keys 'vowels' and 'consonants'
- Call vc_counter() and store results to a dictionary object
- Print the total vowels and consonants with appropriate descriptions (see example).

Notes:

- Only the 26 English letters are counted for this exercise.
- For this exercise, vowels are: A E I O U
- This program does not have to validate or count non-English letters

Hints:

- This will be a lot easier if you make the text file you are evaluating into upper case first
- Don't forget to close the file that you are reading

Example:

```
Enter a text file: War and Peace.txt
Total # of vowels in text file: 1,200,004*
Total # of consonants in text file: 2,900,679*
```

* *counts made up for illustration*

CS-521 Homework Assignment 5

5.2: Write a python program that does the following:

Create 3 functions with docstring:

1. letter_counts() takes a string as its argument and returns a **dictionary of the letters** as keys and frequency counts as values.
2. most_common_letter() takes a string as its argument and returns a list of the most common letter(s). This function should call letter_counts().
3. string_count_histogram() takes a string as its argument and returns a list of the unique letters, with each letter being the repeated number of times it appears in the string. This list will then be printed one element per line (as a histogram).
   This function should call letter_counts().

The following code should be after the functions and inside the block → `if __name__ == '__main__':`

Assign a sentence of at least 15 characters into a constant string variable.
Write 3 print statements with appropriate descriptions that use these functions to create output like the examples.
Notes:

- Line 1 is sorted alphabetically and not printed as a dictionary
- Line 2 grammatically handles more than one result ('appear' vs. 'appears')
- Histogram is sorted alphabetically

Example Output #1

```
The string being analyzed is: "WAS IT A RAT I SAW"
1. 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: "WWWAS IT A RAT I SAW"
1. Letter counts: 'A': 4, 'I': 2, 'R': 1, 'S': 2, 'T': 2, 'W': 4
2. Most frequent letters: 'A', 'W' each appear 4 times.
3. Histogram:
   AAAA
   II
   R
   SS
   TT
   WWWW
```

CS-521 Homework Assignment 5

The following includes concepts taught in Chapter 6

5.3 Write a python program without a function that does the following:

Prompts the user to enter four delimited numbers in one request.
Be sure to tell the user in the prompt what delimiter you are expecting.

Use that input to perform the following calculation:
Multiply the first number by the second number, then add that result to the third number, and finally, divide the result by the fourth number.

This program MUST NOT CRASH.
To insure this you must validate the input data and calculations:

- Checking the user entered four values
- Appropriately checking for the following errors:

ValueError, ZeroDivisionError, IndexError.

  - These must be tested in a single conditional block.
  - *These tests are the key skill being evaluated for this problem*

If there is an error during any validation, the program must:

- Print descriptive error messages to the console.
- Re-prompt the user for the four delimited numbers
  - *There should only be a single input statement for this program.*
  - *The program continues to loop until a valid input is provided*

After generating valid calculation results, print output that shows in one line the formula applied, an equal sign, and the result of the calculation.

Exit the program.

Note:

- Remembering to have a very granular testing block
  - Unrelated commands should not be inside the test
- There can only be one input() in this program

No Example Output provided for this question.

CS-521 Homework Assignment 5

The following includes concepts taught in Chapter 8

5.4: Write a python program that does the following:

Prompt for a file name of text words.
Words can be on many lines with multiple words per line.

Read the file, remove all punctuation (leaving just a space between words), and convert the words to a list.

Call a function you created called list_to_twice_words(), that takes a **list** as an argument and returns a **list** that contains only words that occurred exactly **TWICE** in the file.

Print the results of the function with an appropriate description.

Think about everything you must do when working with a file.

Hint:

- Consider using the string punctuation attribute

No Example Output provided for this question.

CS-521 Homework Assignment 5

The following includes concepts taught in Chapter 15

5.5: This problem is designed for you to practice recursion in a function using factorials.
A factorial is when you multiply a positive integer by the next lower integer all the way down to 1. A factorial is represented as #!. So 5! = 5 * 4 * 3 * 2 * 1 = 120.
Also note that 0! = 1

Write a python program that does the following:

- Prompt the user to enter a factorial starting integer.
  - Make sure it's valid or reprompt the user
- Call your function factorial() that:
  - takes the starting integer as its argument
  - calculated the factorial using recursion
  - returns the value
- Call your function factorial2() that calculates the factorial without recursion
- Print both values with clear descriptions and formatted with thousand commas.

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.