CSC 242: Introduction to Computer Science II

Lab exercise set 1

Due Wednesday, April 1st, 11:30am

Logistics

These exercises should be completed during the lab on Wednesday. In order to receive full credit for the lab, you must attend the session, remain in the lab for at least 45 minutes, and submit a file that contains solutions to all these exercises.

The exercises for this week are a review of material covered in CSC 241. You should find them to be straightforward. If that's not the case, please be sure to review your CSC 241 materials as a part of your first week of work for this class.

You are encouraged to work in groups of **no more than three people** on lab exercises. If you do work with someone, you must include the name(s) of your collaborator(s) at the top of the file you submit. For more information about collaboration policies in this class, see the Academic Integrity policy posted in the syllabus.

If you complete the lab exercise early, please read Chapters 7 and 8 in the textbook and work on the first assignment. You must remain in the lab for at least 45 minutes to earn full credit. You are only allowed to work on assignments with at most two-other people, either directly or indirectly, who must be formally identified as part of your assignment submission. Please see the assignment description and the course Academic Integrity pledge for the full description of the process you must follow when completing assignments. If you need additional help on the assignment, please ask the lab assistant.

Begin the lab by downloading the zip file found on the D2L site. It contains a template file **csc242lab1.py**.

adding the bodies for the functions. When you do, make sure to remove the placeholder pass statements that are currently there.

You must write doc strings for each function. Submissions without doc strings will not receive full credit.

1 Implement a function **longestWord** () that takes one parameter which is a string representing the name of a text file. The function should **return** the length of the largest word in the file. The punctuation in the file should be removed (see str.maketrans) prior to determining the average length of a word in the file. For the purposes of this question the valid punctuation symbols are commas (,), periods (.), exclamation marks (!), question marks (?), semicolons (;), and colons (:). If the file is empty, the function should return 0.

2. Write a function **uniqueWordCount**() that takes as a parameter a string representing the name of a text file. The function **returns** a dictionary of all the unique words in the file and their count. Make sure to remove all punctuation (i.e. all commas, question marks, periods, colons, and semicolons see str.maketrans) before determining whether any words are duplicated so that the punctuation won't interfere with the process**. Capitalization should be ignored**. The information below shows how you would call the function **uniqueWordCount**() and what it would show for a couple of sample executions.

This is what the output looks like when running against the samples files:

```
17
6
0
{'the': 9, 'ba': 2, 'in': 6, 'animation': 10, 'emphasizes': 2, 'principles': 1,
'and': 9, 'production': 2, 'within': 1, 'a': 2, 'broad': 2, 'liberal': 2, 'arts'
: 1, 'context': 1, '2016': 1, 'career': 1, 'review': 1, 'ranked': 1, 'depaul': 1
, '#16': 1, 'program': 2, 'u': 1, 's': 1, 'student': 1, 'will': 1, 'be': 1, 'abl
e': 1, 'to': 4, 'draw': 1, 'creatively': 1, 'innovative': 1, 'connections': 1, '
between': 1, 'other': 1, 'subjects': 1, 'of': 3, 'inspiration': 1, 'this': 2, 'a
rt': 1, 'creativity': 1, 'critical': 1, 'thinking': 1, 'students': 2, 'are': 2,
'introduced': 1, 'foundation': 1, 'skills': 1, 'animating': 1, 'then': 1, 'take'
: 1, 'sequence': 1, 'which': 1, 'they': 1, 'expected': 1, 'bring': 1, 'intellect
ual': 1, 'creative': 1, 'viewpoint': 1, 'influenced': 1, 'by': 1, 'their': 2, 's
tudies': 1, 'courses': 1, 'electives': 1, 'work': 1, 'can': 1, 'experiment': 1,
'with': 2, 'latest': 1, 'software': 1, 'tools': 1, 'technologies': 1, 'including
': 1, 'hand-drawn': 1, '3d': 1, 'stop': 1, 'motion': 3, 'capture': 1, 'control':
 1, 'green': 1, 'screen': 1, 'studios': 1, 'is': 1, 'an': 2, 'extremely': 1, 'fl
exible': 1, 'degree': 1, 'emphasis': 1, 'on': 1, 'interdisciplinary': 1, 'learni
ng': 1, 'self-directed': 1, 'exploration': 1}
{'this': 1, 'is': 1, 'a': 1, 'simple': 1, 'doc': 1, 'to': 1, 'test': 1}
{}
```

Submitting the exercises

Modify the lab1.py template file provided on D2L and upload it to the associated D2L dropbox.

Grading

The lab session is worth 10 points. See the lab grading rubric for more detail.

Submitting the exercises

You must submit your solution to the exercises using the lab 1 dropbox on the D2L site. Submit only a single text file (**csc242lab1.py**) with each of the completed functions and classes for the lab exercises in it. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The lab session is worth 10 points. If you complete the lab exercises before the end of the lab session, please work on the first assignment. Remember that the rules for collaboration on assignments is different from labs. Please review the Academic Integrity policy for more information. If you have questions about the assignment, please ask the teaching assistant.