Project 02 - Using Modules with Character Counting

Due Feb 5 by 11:59pm **Points** 50 **Submitting** a file upload

Overview

In this project we will be adding to the character counting program from Project 1. We will modify it to use modules, write to csv files, and we will test thoroughly with the unittest module.

What to do

1. Modularize

Convert your program to a module. At the bottom of the file, put in the necessary code to call the main function if the code is not being imported. Now that it can be imported as a module without executing all the code, break out the core functionality that used to be in main() of this program into a separate function. You may modify this function so that it returns the dictionary rather than printing it.

Essentially make sure your functions can be run from the testing .py file so you can test each flag option. This is all this section is asking for.

2. Write to csv file

In a separate file called count_to_csv.py, import your count.py. This script will take an additional argument over count.py: the final argument should be the name of a csv file to which you will write the data printed. This file does not need to already exist.

For example:

python3 count_to_csv.py -z -c test1.txt test2.txt out.csv

3. Unit test

Create another file, test_count.py. This should contain the unit testing code for count.py. You do not need to unit test count_to_csv.py. Make sure to write test cases for every combination of flags.

Hint: sys.argv is mutable.

Note: The Unit Tests will likely run in parallel all at once, which will sometimes cause issues. I recommend first checking each test individually and seeing if they pass as expected

Submission

Push your code to your git repository, inside a folder called Project2 and also upload the files

Project 02 - Counting Characters Extended- Rubric

Criteria	Ratings	Pts
Coding Style Comments are used well to explain the code. Variables are named reasonably. Indentation matches expectations. Named constants are used instead of placing values directly into the code. Code is generally readable. Etc.		8 pts
Modularization count.py can be run in a testable way		10 pts
count_to_csv count_to_csv.py correctly imports and uses count.py. It correctly writes csv format to a file. The filename it is written to is correctly parsed as an argument from the command line.		10 pts
test_count.py test_count.py contains sufficient test cases, and the test cases describe correct behavior.		20 pts
Uploading To Git The project is uploaded correctly to your git repository		2 pts
		Total Points: 50