DAT100 – Foundations of Computer Programming

Final Test – Spring 2024

Instructions:

This test is open book. Also, you may use IDLE (or another Python IDE of your choice) to test your code. However, you may not discuss the questions with anyone else.

You will be required to create code in programs (scripts) and modules, and you will need to submit them to the Assignment folder for this test.

You have until the test is no longer available (dates on A2L) to complete the test.

Test questions (35 marks)

Please complete the following tasks and put all the files into one zip folder and submit it to the Final Test folder in the Assignments area.

1. Create a program called CreateDataFile() with the following requirements:
    a. The user is prompted for a filename to write data to. You must make sure the user enter a filename. For the purposes of this test, a valid filename is any string except an empty string. This filename is the file you will create (or overwrite) in the program specified in parts b. and c. of this question.
    b. The user enters random string of characters as data. The data is entered one string (one line) at a time in a loop. Data entry is complete when the user presses the Enter key with no data (an empty string).
    c. As each string is entered by the user, it is written to the file that was specified by the user in part a. above. Each string entered is on a new line in the file. There must be no blank lines in the file.
    d. Save this program in a file called CreateDataFile.py

    (8 marks)

2. Create a *function* called FileRead() with the following requirements:
    a. The function uses a single parameter with the name of the text file to open and read.
    b. Each line of the text file must be read and added to a Dictionary. In each element of the dictionary, the key is the line number read from the file and the value is the string read from the file. Convert each line read to lower case.
    c. The return value of the function is the Dictionary that was created while reading the file.
    d. Save this *function* in a file called FileRead.py
    e. Create a separate program to test this function, making sure you import the function module. Name the program TestFileRead.py
    You may use the file created in question 1 as a test file.

    (8 marks)

(Continued on next page)

3. Create a function called GetValue() with the following requirements:
    a. Two parameters
        i. the first parameter is the name of Dictionary with the same structure as the one created in question 2 above (i.e. the key is an integer and the value is a string)
        ii. the second parameter is the key of the value you want to retrieve
    b. The return value is the value retrieved from the Dictionary.
    If the operation failed, then return the string "!!!None!!!".
    c. Save this function in a file called GetValue.py
    d. Create a separate program to test this function, making sure you import the function module. Name the program TestGetValue.py
    You may use the Dictionary created in question 2 in the test program.

(9 marks)

4. The following function creates a random number from 1000 to 5000 (inclusive):

```
import random
def getRandom():
    val = int(random.random()  * 4000) + 1000
    return val
```

    a. Create a module with this single function (including the "import random" statement) in a file called GetRandom.py
    b. Create a program that does the following:
        i. Generate two **sets** of 1500 unique elements, each a random integer between 1000 and 5000 inclusive. You must demonstrate the use of the getRandom() function by importing the module GetRandom into this program.
        ii. Display the number of elements that duplicated in both sets. (You **must** use **set operators/methods** as part of your solution to this problem.)
        Do NOT display the elements themselves in your final solution – just the number of elements that are the same.
        iii. Call this program RandomSetAnalysis.py

(10 marks)

Please put all the files (7 of them, not including any data files) into one zip folder and submit it to the Final Test folder in the Assignments area.

(End of test)