Individual Programming 3 (Part 2)

Start Assignment

Due Saturday by 1pm **Points** 17 **Submitting** a file upload **File Types** py, docx, doc, txt, and csv
Available Oct 30 at 1:30pm - Nov 13 at 1:30pm 14 days

This is an individual programming assignment. Although you are welcome to collaborate and exchange ideas with others, all work submitted for this assignment should be your own. Note, all individual programming assignments only require ideas presented in the quizzes, textbook, and lecture. You should not need to do additional research to complete the main task (though you are welcome to do so).

The following task is worth 12 points. For the remaining 5 points, see "extension" further below.

Write a program that creates a dictionary from a vocabulary list (formatted in a file similar to **vocablist.txt** (https://my.uclaextension.edu/courses/36536/files/5638396/download?download_frd=1) and **myvocab.txt** (https://my.uclaextension.edu/courses/36536/files/5638407/download?download_frd=1)). If the user-entered file name does not exist, the program should ***catch*** the error and re-prompt for a file name.

After the file is read in successfully, the program should use the "term" as the key and "definition" as the value. The program should then ask the user if he/she would like to add additional terms and definitions, and how many if "Y" is the response. If the user enters a *negative* non-integer value, the program will prompt the user again for an integer value. The program should then prompt the user for as many terms and definitions as needed.

The program should ask again if the user would like to add additional terms and definitions. Again, the program will ask "how many" if "Y" is the response. The program should loop between asking if the user would like to add additional terms and definitions, and how many, until the user types "N" for No.

If a duplicate term (key) is entered, the program will notify the user that the term is already in the dictionary. It will ask if the user would like to update the definition BEFORE asking for a definition.

The program should end by printing the list of terms and definitions before asking the user for a file name to save the new dictionary. **DO NOT SAVE OVER THE ORIGINAL TEXT FILE UNLESS ENTERED BY THE USER.** You may assume that the file will be a .txt file (with text separated by tabs "\t").

Below, you will find examples of what the program might look like, with red as the user input:

```
##Example 1
Please enter a file name: vocablist.txt
There are 10 terms in this vocabulary list. Would you like to add more terms (Y/N)? Y
How many would you like to add? 2
Term #1: IDLE
Definition #1: Integrated development environment
Term #2: if
Definition #2: Conditionally executes a block of code
Would you like to add more terms (Y/N)? Y
How many would you like to add? 0
Would you like to add more terms (Y/N)? N
There are 12 terms in the new vocabulary list.

break - Used to exit a for loop or a while loop.
continue - Used to skip the current block, and return to the "for" or "while" statement
dictionary - A mutable associative array (or dictionary) of key and value pairs.
float - An immutable floating point number.
immutable - Cannot be changed after its created.
int - An immutable integer of unlimited magnitude.
pass - Needed to create an empty code block
set - Unordered set, contains no duplicates
string - Can include numbers, letters, and various symbols and be enclosed by either double or single quotes
while - Executes a block of code as long as its condition is true.
IDLE - Integrated development environment
if - Conditionally executes a block of code
```

```
What would you like to save the file as? vocablist2.txt


##Example 2
Please enter a file name: vocablist.txt
There are 10 terms in this vocabulary list. Would you like to add more (Y/N)? Y
How many would you like to add? 1.5
Error. Please enter an integer.
How many would you like to add? -4
Error. Please enter a positive number.
How many would you like to add? one
Error. Please enter an integer.
How many would you like to add? 2
Term #1: break
Warning! This term is already in the vocabulary list. Update definition (Y/N)? N
Term #2: if
Definition #2: Conditionally executes a block of code
Would you like to add more terms (Y/N)? Y
How many would you like to add? 0
Would you like to add more terms (Y/N)? N
There are 11 terms in the new vocabulary list.

break - Used to exit a for loop or a while loop.
continue - Used to skip the current block, and return to the "for" or "while" statement
dictionary - A mutable associative array (or dictionary) of key and value pairs.
float - An immutable floating point number.
immutable - Cannot be changed after its created.
int - An immutable integer of unlimited magnitude.
pass - Needed to create an empty code block
set - Unordered set, contains no duplicates
string - Can include numbers, letters, and various symbols and be enclosed by either double or single quotes
while - Executes a block of code as long as its condition is true.
if - Conditionally executes a block of code

What would you like to save the file as? vocabulary.txt


##Example 3
Please enter a file name: vocablist.txt
There are 10 terms in this vocabulary list. Would you like to add more (Y/N)? Y
How many would you like to add? 1.5
Error. Please enter an integer.
How many would you like to add? -4
Error. Please enter a positive number.
How many would you like to add? one
Error. Please enter an integer.
How many would you like to add? 2
Term #1: break
Warning! This term is already in the vocabulary list. Update definition (Y/N)? Y
Definition #1: Used to exit a loop
Term #2: if
Definition #2: Conditionally executes a block of code
Would you like to add more terms (Y/N)? Y
How many would you like to add? 0
Would you like to add more terms (Y/N)? N
There are 11 terms in the new vocabulary list.

break - Used to exit a loop
continue - Used to skip the current block, and return to the "for" or "while" statement
dictionary - A mutable associative array (or dictionary) of key and value pairs.
float - An immutable floating point number.
immutable - Cannot be changed after its created.
int - An immutable integer of unlimited magnitude.
pass - Needed to create an empty code block
set - Unordered set, contains no duplicates
string - Can include numbers, letters, and various symbols and be enclosed by either double or single quotes
while - Executes a block of code as long as its condition is true.
if - Conditionally executes a block of code

What would you like to save the file as? mynewvocablist.txt
```

EXTENSION (5 points):

Note: You cannot earn credit for this portion if you are submitting your assignment past 11/8 at 11:59pm.

This is your own customization of the original assignment. Think of 3+ that you can make the current program more flexible or user-friendly. These changes cannot be coding efficiency improvements (as your original program should already be at its best). The changes are open-ended and MUST be visible to someone using your program. For some improvements, it might help to look in the upcoming chapters. For the "extension" portion of this assignment, you will need to download and complete this "Programming Extension Form": **extension_template.docx** (https://my.uclaextension.edu/courses/36536/files/5638400/download?download_frd=1)

In all, you should submit three files:

1. **Original assignment exactly as prompted above (.py file)**
2. **Programming Extension Form (.docx file)** (https://my.uclaextension.edu/courses/36536/files/5638400/download?download_frd=1)
3. **Modified .py file with changes as listed in the Extension Form**

IndividualProgramming

Criteria	Ratings			Pts
Task	**6 pts** **Full Marks** Fully addresses all required elements of the task, without error. Program runs EXACTLY as in the examples, without deviations in text (including spaces or punctuations). Original vocabulary text file is preserved (not modified).	**3 pts** **Half Marks** Addresses almost all of the required elements of the task. The program runs, but the saved file has some error or there are some deviations in the program text.	**0 pts** **No Marks**	6 pts
Functions	**2 pts** **Full Marks** Programmer appropriately uses functions to handle small tasks with an underlying idea or concept		**0 pts** **No Marks**	2 pts
Efficiency	**2 pts** **Full Marks** Programmer uses recent concepts to handle tasks efficiently and with minimal coding. Code is free of redundancy.		**0 pts** **No Marks**	2 pts
Readability	**2 pts** **Full Marks** Code is well-commented and easy to follow. Variables are appropriately named.		**0 pts** **No Marks**	2 pts
Extension	**5 pts** **Exceptional** The programmer submits both a original .py file and an extension .py file. It is clear in the header file which file is the original and which file is the extension. The programmer also submits an accompanying "Program Extension Form" clearly detailing changes (at least 3 that are visible from the user's point of view) to the programs, and why the changes make the extension program even better than the original. The changes are exceptional and well-tested for a variety of user-inputs.	**3 pts** **Above Average** The programmer submits both a original .py file and an extension .py file. It is clear in the header file which file is the original and which file is the extension. The programmer also submits an accompanying "Program Extension Form" detailing changes to the programs, but it is not clear why the changes make the extension program even better than the original. There about 2-3 changes to the original program, and the program can handle a variety of user-inputs.	**1 pts** **Meets Extension Requirement** The programmer submits both a original .py file and an extension .py file. It is clear in the header file which file is the original and which file is the extension. The programmer also submits an accompanying "Program Extension Form" detailing changes to the programs, but it is not clear why the changes make the extension program even better than the original. There is only one minor change to the original program.	**0 pts** **No Marks**	5 pts
				Total Points: 17