



PYTHON

PROGRAMMING-PILOT

(355)

REGIONAL – 2022

**PRODUCTION PORTION:**

Program 1: Word List Reader/Writer (250 points)

***TOTAL POINTS***  *(250 points)*

**GENERAL GUIDELINES:**

*Failure to adhere to any of the following rules will result in disqualification:*

1. Contestant must hand in this test booklet and all printouts if any. Failure to do so will result in disqualification.
2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests (handwritten, photocopied, or keyed) are allowed in the testing area.
3. Electronic devices will be monitored according to ACT standards.

*You will have ninety (90) minutes to complete your work.*

Your name and/or school name should *not* appear on work you submit for scoring.

1. Create a folder on the flash drive provided using your contestant number as the name of the folder.
2. Copy your entire solution/project into this folder.
3. Submit your entire solution/project so that the graders may open your project to review the source code.
4. Ensure that the files required to run your program are present and will execute on the flash drive provided.

\*Note that the flash drive letter may *not* be the same when the program is scored as it was when you created the program.

\*It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter.

The graders will *not* compile or alter your source code to correct for this.   
Submissions that do *not* contain source code will *not* be scored.

**Assumptions to make when taking this assessment:**

* The user will *not* enter invalid input.
* The input file will only contain valid input strings that only contain upper and lower case letters.

**Development Standards:**

* Your Code must use a consistent variable naming convention.
* All functions (if any) must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any).
* If you create a class, then you must document the class and its methods.

**Word List Reader/Writer**

As an entry-level programmer, you are tasked with creating a Python application to store a list of words and then read the list to determine characteristics of the word set.

Task 1: Write a program that asks the user how many words they would like to write to a file, and then asks the user to enter that many words, one at a time. The words should be written to the file named “words.txt”.

Task 2: Add the following functionality to your program to read the words from the file “words.txt” and display the following:

* The number of words in the file.
* The longest word in the file.
* The average length of all of the words in the file.

**Requirements:**

1. You must create an application called WordList.
2. Your program must ask the user how many words they would like to write to a file.
3. Your program must ask for the correct number of words to write.
4. Your program must storage the words (one per line) to the file named “words.txt”.
5. Your program must calculate the number of words in the file. (NOTE: Assume the judges will modify the content in the file named “words.txt” to test the function).
6. Your program must calculate the longest word in the file. (NOTE: Assume the judges will modify the content in the file named “words.txt” to test the function).
7. Your program must calculate the average length of the words in the file. (NOTE: Assume the judges will modify the content in the file named “words.txt” to test the function).
8. Your contestant number must appear as a comment at the top of the main source code file.
9. Your program must be commented appropriately.

|  |  |  |
| --- | --- | --- |
| **Solution and Project** |  | |
| The project is present on the flash drive |  | 10 points |
| The project is named WordList |  | 10 points |
|  |  |  |
| **Program Execution** |  |  |
| The program runs/compiles from the USB flash drive |  | 10 points |
| ***If the program does not execute, then the remaining items in this section receive a score of zero.*** | | |
| The program prompts the user for a specific number of words to enter. |  | 10 points |
| The program asks for and limits the number of words to enter based on the initial prompt. |  | 10 points |
| The program stores each word (one per line) in the file named “words.txt”. |  | 10 points |
| The program reads the words from the file named “words.txt”. |  | 10 points |
| The program properly calculates the number of words in the file named “words.txt” and displays it to the user. |  | 10 points |
| The program properly calculates the longest word in the file named “words.txt” and displays it to the user. |  | 10 points |
| The program properly calculates the average length of the words in the file named “words.txt” and displays it to the user. |  | 10 points |
|  |  |  |
| **Source Code Review** |  |  |
| A comment containing the contestant number is present |  | 10 points |
| Functions and code sections are commented |  | 30 points |
| Code is present to prompt the user for a number of words. |  | 10 points |
| Code is present to obtain the words one by one up to the requested number of words. |  | 10 points |
| Code is present to read the words in the file named “words.txt”. |  | 10 points |
| Code is present to calculate the number of words in the file named “words.txt” and display the output to the user. |  | 20 points |
| Code is present to calculate the longest word in the file named “words.txt” and display the output to the user. |  | 20 points |
| Code is present to calculate the average word length in the file named “words.txt” and display the output to the user. |  | 20 points |
| All files are opened and closed properly |  | 10 points |
| Code uses a consistent variable naming convention |  | 10 points |
|  |  |  |
| **Total Points =\_\_\_\_ / 250 points** | | |

SAMPLE SOURCE CODE

# The main function.

def main():

# Get the number of words to write to the file.

word\_count = int(input('Enter number of words to write: '))

# Open output file.

outfile = open('words.txt', 'w')

# Write specified number of random numbers to the file.

for counter in range(1, word\_count + 1):

# Get a word.

print('Word', counter, 'of', word\_count, '- ', end='')

word = input('Enter a word: ')

# Write the word to the file.

outfile.write(word + '\n')

# Close the file.

outfile.close()

# Local variables

counter = 0

total\_length = 0

longest\_word = ''

# Open input file

infile = open('words.txt', 'r')

# Read numbers from the file while keeping count

# and a running total

for word in infile:

# Strip line break from end of word and get its length.

word = word.rstrip('\n')

word\_length = len(word)

# Add length to total length and increment the counter.

total\_length += word\_length

counter += 1

# Check for longest word.

if word\_length > len(longest\_word):

longest\_word = word

# Close the file.

infile.close()

# Determine average length.

average\_length = total\_length / counter

# Display the results.

print('Number of Words:', counter)

print('Longest Word:', longest\_word)

print('Average Word Length:', round(average\_length))

# Call the main function.

main()