CITC-1301 Introduction to Programming

Chapter 8 Lab – Pig Latin Translator

Write a Python program that translates the contents of a text file to Pig Latin. The program shall ask the user for a file, translate all the words in the file to Pig Latin maintaining its syntax (e.g., periods, commas, etc.), and writes the translation to a new file.

Use the following rules to translate words to Pig Latin:

- For works beginning with a vowel: append "*yay*" to the end of the word.
- For words beginning with a consonant: move the first letter of the word to the end of the word and append "*ay*" at the end of the word.

Other rules:

- Maintain proper capitalizations (e.g., "*Three*" is translated as "*Hreetay*" – with the "H" capitalized).
- Maintain phrase and sentence terminators (e.g., the phrase ending with "*stone,*" is translated as "*tonesay,*" – with the "," moved to the end of the translated word.
  - Do the same for commas, colons, semicolons, question marks, and explanation points.
- Maintain the original locations of apostrophes and dashes.
- Do not translate numeric values (e.g., "*17*" should translate to "*17*").

Fixing existing code

Copy and paste the starting code on pages 2 through 3 into your editor. Most of the code for this program is already written. You only have to write the implementation for the **`translateWordToPigLatin()`** function.

The **`translateWordToPigLatin()`** function accepts one argument: a **`word`**. Based on the rules enumerated above, translate the passed-in **`word`** into Pig Latin and return the resulting translation to the function-call.

> Hint: It might be easier to write and test the **`translateWordToPigLatin()`** function in a separate solution file. Structured error-handling has been implemented throughout the program; it may be difficult to debug the function otherwise.

Do not change any of the other code.

Example output:

```
Pig Latin Translator

This program translates the contents of a text file to Pig Latin.

File path: rings.txt [ENTER]

Translating...
Translated file written to ringsPigLatin.txt
```

Example input file (rings.txt):

```
Three Rings for the Elven-kings under the sky,
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie.
One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie.
```

Example output file (ringsPigLatin.txt):

```
Hreetay Ingsray orfay hetay Elven-kingsyay underyay hetay kysay,
Evensay orfay hetay Warf-lordsday inyay heirtay allshay ofyay tonesay,
Inenay orfay Ortalmay Enmay oomedday otay ieday,
Oneyay orfay hetay Arkday Ordlay onyay ishay arkday hronetay
Inyay hetay Andlay ofyay Ordormay hereway hetay Hadowssay ielay.
Oneyay Ingray otay uleray hemtay allyay, Oneyay Ingray otay indfay hemtay,
Oneyay Ingray otay ringbay hemtay allyay andyay inyay hetay arknessday indbay hemtay
Inyay hetay Andlay ofyay Ordormay hereway hetay Hadowssay ielay.
```

Starting code:

```python
# Chapter 8 Lab - Pig Latin Translator


def main():
    print("Pig Latin Translator\n")
    print("This program translates the contents of a text file to Pig Latin.\n")

    while True:
        try:
            # Get file path from user
            filePath = input("File path: ")

            # Attempt to open and read contents of specified file and
            # Split the text into a list based on newline character (each line of text becomes an element in list)
            linesList = readFile(filePath).split("\n")
        except Exception as ex:
            # If there is an error opening file
            print(ex)
        else:
            break       # Exit loop if there are no problems
        # end try
    # end while

    # Translate and add each line to translated list
    translatedLinesList = []
    for line in linesList:
        translatedLinesList.append(translateLineToPigLatin(line))
    # end for

    try:
        # Determine name for translated file
        periodIndex = filePath.rfind(".")
        translatedFilePath = filePath[:periodIndex] + "PigLatin" + filePath[periodIndex:]

        # Write translation to file
        print("\nTranslating...")
        writeFile(translatedFilePath, translatedLinesList)
        print("Translated file written to", translatedFilePath)
    except Exception as ex:
        # If there is an error writing file
        print(ex)
    # end try
# end function
```

```
def readFile(filePath):
    try:
        # Open file path for reading
        fileObject = open(filePath, "r")

        # Read all lines from file
        fileContents = fileObject.read()
    except FileNotFoundError:
        raise FileNotFoundError("Unable to open file: the specified file does not exist.")
    except:
        raise Exception("An unexpected problem occurred whilst opening the file.")
    else:
        fileObject.close()

        return fileContents
    # end try
# end function


def writeFile(filePath, linesList):
    try:
        # Open file path for writing
        fileObject = open(filePath, "w")

        # Ensure there is at least 1 item in list
        if len(linesList) > 0:
            # Process each line of text in list
            for line in linesList:
                # Write each translated line to file
                fileObject.write(line)
            # end for
        else:
            raise Exception("Unable to translate: the file does not contain any text.")
        # end if
    except:
        raise Exception("An unexpected problem occurred whilst writing the file.")
    else:
        fileObject.close()
    # end try
# end function


def translateLineToPigLatin(line):
    translatedLine = ""

    # If line has one or more character
    if len(line) > 0:
        # If line ends with a newline character, remove it
        if line.endswith("\n"):
            line = line.rstrip("\n")
        # end if

        # Split line into a words list based on a space
        wordList = line.split(" ")

        # Concatenate each translated word with a trailing space to translatedLine
        for word in wordList:
            translatedLine += translateWordToPigLatin(word) + " "
        # end for
    # end if

    return translatedLine + "\n"    # Return translated line with trailing newline character
# end function


def translateWordToPigLatin(word):
    return ""    # TO DO
# end function


# Call main function
main()
```

Submission Instructions

- Upload your Python script (i.e., your .py file) to the appropriate dropbox on D2L.