**CIS 229 – Python Programming – Programming Assignment**

**Genesis Alphabetizer**

Overview

In this assignment, you will be writing a program that analyzes a text file that contains the book of Genesis from the King James version of the bible (data file included with assignment). Ultimately, your program will generate a text file that contains an alphabetized list of every unique (non-repeating) word that appears in Genesis.

When completing this assignment, the student should demonstrate mastery of the following concepts:

* Lists
* Iterating By List
* String Manipulation
* String Evaluation
* Reading From File
* Writing To File

Assignment

Your program should begin by reading each line of text from the file and storing it in a string. Python contains a useful utility function called split() that can be used to break a string that contains many words into list of words. Consider the following code to understand how split works:

def main():

# make a string that contains multiple words

manyWordString = "This is a string that has many words in it."

print("Before Splitting:\n{}\n".format(manyWordString))

# split the string into a list of words

wordList = manyWordString.split()

# iterate through the wordList with an index to

# visualize what the split function did

print("After Splitting:")

for index in range( len(wordList) ):

print("wordList[{}] contains: {}".format(index, wordList[index]))

main()

Split Program Output

Before Splitting:

This is a string that has many words in it.

After Splitting:

wordList[0] contains: This

wordList[1] contains: is

wordList[2] contains: a

wordList[3] contains: string

wordList[4] contains: that

wordList[5] contains: has

wordList[6] contains: many

wordList[7] contains: words

wordList[8] contains: in

wordList[9] contains: it.

After using split, each word (space delimited token) will be an element in the list for a given line of the text file. From there, the candidate word must be purified before it can be tested for uniqueness and cataloged. In order for a word to be cataloged, it must contain only letters (no punctuation). Additionally, we will convert all of the letter in the word to uppercase so we can later use the sort() method for lists to put the words in alphabetical order.

To purify the word, the string within the list will need to be analyzed on a character by character basis and a new string that contains only letters should be constructed (isalpha() would be useful for this). After that, convert the string to all upper-case letters. From there, each purified string can be appended (use the append() method for lists) to a master list after it has been checked for uniqueness against all of the other words that have already been put in the list. Finally, make a call to the sort() method for lists and write the alphabetized list into an output file. A sample of a valid output file is also included with this assignment so you can check your results.

Assessment

This assignment will be assessed based on the provided grading rubric.