Reading Data

POTD 15 (http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/)

My answer:

```
# Write a function that takes a sentence,
# check if there are repeated words.
# For each repeated word, keep track the number of time it repeats.
# Your program will return a collection of the repeated words and
# the number of the repetition.
# Let's assume we want to use a dictionary as a data type.
# Thus, a key-value pair would be
#       word:number
#
# If there is no repeated word or the file is empty, return an empty dictionary
#  -- that is, {}
#
# For example, if a sentence contains
#      'I will do more more practice and my bring my my my questions to class'
#
# Your program will return
#      {'more':1, 'my':2}    # 'more' repeats 1 time, 'my' repeats 2 times

def check_repeated_words(sentence):
    result = {}
    sentence = sentence.lower()
    sentence = sentence.replace(',', ' ')
    sentence = sentence.replace('?', ' ')
    sentence = sentence.replace('!', ' ')
    sentence = sentence.replace('.', ' ')
    sentence = sentence.replace(':', ' ')
    sentence = sentence.replace(';', ' ')
    lst = sentence.split()
    for n in range(len(lst) - 1):
        if lst[n] == lst[n+1]:
            if lst[n] in result:
                result[lst[n]] += 1
            else:
                result[lst[n]] = 1
    return result


print(check_repeated_words('I will do more more practice and my bring my my my questions
to class'))
# prints {'more': 1, 'my': 2}
print(check_repeated_words("This this this this; sentence! sentence exists to to to to.
to. to, to to, test test test my my code code to code"))
# prints {'this': 3, 'sentence': 1, 'to': 7, 'test': 2, 'my': 1, 'code': 1}
```

***class answer was written on the blackboard, but used the same strategy to solve the problem

***the replaces and the .lower() aren't necessary for the given input, but I wanted to make it work for more inputs

Opening a local file: **file_variable = open(filename, mode)**

Modes are: **'r'** read-only, **'w'** write (like 'enable editing' in word), **'a'** append (can't change what's there, but can add to the end)

If the file we are accessing isn't in the same directory as the python file we are writing, we can access it by including a "path" in its name

This "path" includes the folder the file is in, the folder that folder is in, and so on until it reaches the C: drive

Because Archimedes doesn't have access to our C: drive, homeworks must be submitted with the file in the same directory

To select a line from the file, **file_variable.readline()**

To read an entire file, iterate readline

To read the whole thing without line breaks, **file_variable.read()**

To write in a file, open the file, then **file_variable.write('string we want written')**

Will automatically start writing at the end of the line

When done with a file, we need to close it: **file_variable.close()**

If we open a file in two places and edit it in one, we get inconsistencies (just like a Word document); closing updates the official version of the file

In-class example:

Text file created:

```
Layne
Stephanie
Marie
Marilyn
Maria
Mary
Kyle
Anjali
Craig
Luther
Keven
Kevyn
Kevon
Kevin
```

Python file created:

```
# open friends.txt
# read it
# add @virginia.edu to each name
# write name, email to emails.txt

def read_file(filename):
```

```
    '''
    returns list of names
    '''
    result = []
    infile = open(filename, "r")
    # infile stands for input file
    # similarly, outfile stands for output file
    for line in infile:
        # print(line.strip())
        # .strip() removes "space characters" i.e. spaces, tabs, and /n s
        # the file from debugging practice included a strip command with more things to
remove
        result.append(line.strip() + ", " + line.strip() + "@virginia.edu")
    return result

def write_file(list_of_data):
    outfile = open('emails.txt', 'a')
    for line in list_of_data:
        outfile.write(line + '\n')

list_of_friends = read_file("friends.txt")
write_file(list_of_friends)
```

Text file created by the Python file:

```
Layne, Layne@virginia.edu
Stephanie, Stephanie@virginia.edu
Marie, Marie@virginia.edu
Marilyn, Marilyn@virginia.edu
Maria, Maria@virginia.edu
Mary, Mary@virginia.edu
Kyle, Kyle@virginia.edu
Anjali, Anjali@virginia.edu
Craig, Craig@virginia.edu
Luther, Luther@virginia.edu
Keven, Keven@virginia.edu
Kevyn, Kevyn@virginia.edu
Kevon, Kevon@virginia.edu
Kevin, Kevin@virginia.edu
```