Lab 3
Python File Handling

Part 1 - read entire file at once

Get the data file sample.txt from your instructor and place a copy of it in the same folder you have been saving your Mu python files.

Copy the code listed below in to the Mu editor and save your code with the filename “**lastname_lab3_part1.py**”

```
#
fileobject = open("sample.txt", "r")
#
filedata = fileobject.readlines()
#
print(len(filedata), " lines in the file!")

#
count=0
#
for line in filedata:
    #
    word = line.split()
    #
    count = count + len(word)

#
print("\n There are "+str(count)+" words in the file")
#
file.close()
```

Study the code and figure out what it is doing. Add your header comment to the top of the file and add inline comments after every # in the program.

Open a text editor and create another text file in the same directory. Use the program to read the data from that file as well.

What type of data structure is `filedata` and what type of data does it contain? Use the `type()` function to confirm your guesses.

filedata is a list of strings
Solutions file is lab3_part1.py

Part 2 - read a file line by line

Copy the code listed below into the Mu editor and save your code with the filename “**lastname_lab3_part2.py**” Add your header and inline comments to the file.

```
#
count = 0
#
with open("sample.txt") as fileobject:
#
    for line in fileobject:
#
        words = line.split()
#
        count = count + len(words)
#
print("There are "+str(count)+" words in the file")
```

Solutions file is lab3_part2.py

What is the data type of the `line` variable? line is a string

Compare and contrast the code from part 1 and part 2. What might be a benefit of reading file data as it was done in part 1? What might be a benefit of reading data from a file as it was done in part 2?

Part 1 you get ALL the data as a list of strings. If you need to compare one line to another line of data in the file this may be the way to go. Part 2 you get individual strings. If you know you are going to process each line the same way, this may be the way to go and not bog down memory resources by reading the entire file.

You should consider how much of the file you need to access at a time and what methods are best suited for your needs.

FYI there is also a .read() method.

```
#The difference between file.read(), file.readline(), file.readlines()
file = open('samplefile', 'r')
single_string = file.read() #Reads all the elements of the file #into a single string(\n characters might be included)

line = file.readline() #Reads the current line where the cursor as a string #is positioned and moves to the next line

list_strings = file.readlines()#Makes a list of strings
```

Part 3 - write to a file/ append to a file

Copy the code listed below into the Mu editor and save your code with the filename "**lastname_lab3_part3.py**"

```
#
my_file_obj = open("coolfile.txt","w")
#
my_file_obj.write("cheeseburgers are good\n")
#
my_file_obj.write("hotdogs are ok too\n")
#
my_file_obj.close()
```

Open the resulting file, "coolfile.txt", in a text editor to view the contents.
The file coolfile.txt should contain the 2 strings written. WRITE mode
Change the "w" in your code to "a" and run the program a few times. What happens to the resulting output file?
The file coolfile.txt should contain the 2 strings written over and over again, as many times as the program was run. APPEND mode
Change the "a" back to "w" and run the program. What happened to the output file?
The file coolfile.txt should contain the 2 strings written. WRITE mode deletes previous contents of the file. APPEND mode keeps any previous file contents and appends to the end of the file.
Add comments to the file.
Solutions file is lab3_part3.py

Part 4 - add line numbers to a file

Create a new file in the Mu editor and save it as "**lastname_lab3_part4.py**".
Write a program that will open "sample.txt", read each line from the file, insert line numbers at the beginning of every line and write the numbered lines into a new file called "numbered_sample.txt"

E.g. if sample.txt was:

Monkey funky
Dog fog
Cat Fat
Mouse House

numbered_sample.txt would become:

1 Monkey funky
2 Dog fog

3 Cat Fat
4 Mouse House

Make sure to add comments to the file.

Possible solutions file is lab3_part4.py

- Use the Mu debugger to step through the code to help find problems.