

hdd 249 Dan Schumacher Week 3 Lab Exercises

Blank notebook to be used for class exercises.

Exercise 1

Read the file "mbox.txt" line-by-line and calculate and print the following:

- The total number of lines in the file.
- The number of lines that contain the substring "From:"

```
In [1]: #problem 1
file_handle = open('mbox.txt')
count = 0
for line in file_handle:
    count+=1
print('Total lines:', count)

#problem 2
file_handle = open('mbox.txt')
count = 0
for line in file_handle:
    if 'From:' in line:
        count+=1

#return results and close handle
print('"From" lines: ', count)
file_handle.close()
```

```
Total lines: 132045
"From" lines: 1797
```

Exercise 2

Write code that reads the file line-by-line numbers.txt, then does the following:

- Sum all the numbers in numbers.txt, then prints the numbers to the screen.

Next, append the string "SUM: k", where k is the calculated sum. to the end of numbers.txt as a new line.

Finally, print the entire file to make sure you appended to the file correctly.

The absolute path for the file is "../data/numbers.txt"

Important Python concepts: for, open ('a' and 'r'), write(), print()

```
In [2]: # If you mess up the file and need to try again, run this cell again to recreate the f
with open('./numbers.txt', 'w') as out_file:
    out_file.write('42\n18\n22\n18')

with open('numbers.txt', 'r') as in_file:
    for row in in_file:
        print(row.strip())
```

```
42
18
22
18
```

```
In [3]: file_handle = open('numbers.txt', 'r')
score = 0
for line in file_handle:
    score += int(line.strip())
file_handle.close()

file_handle = open('numbers.txt', 'a')
file_handle.write(f'\nSUM: {score}')
file_handle.close()

with open('numbers.txt', 'r') as in_file:
    for row in in_file:
        print(row.strip())
file_handle.close()
```

```
42
18
22
18
SUM: 100
```

Run the line below to check your work

```
In [4]: with open('numbers.txt', 'r') as in_file:
        for row in in_file:
            print(row.strip())
```

```
42
18
22
18
SUM: 100
```

Exercise 3

Write a function to calculate your pay given two arguments: hoursWorks and dollarsPerHour. The function should return how much you should be paid. When calculating the final amount, give the employee 1.5 times the hourly rate for hours worked above 40 hours.

```
In [5]: # Test arguments
hours = 45 # You worked for 45 hours
pay = 10 # You make 10 dollars and hour
```

```
In [6]: def payday(hoursWorks, dollarsPerHour):
        overTime = 0
        if hoursWorks > 40:
            overTime = hoursWorks-40
            hoursWorks = 40
        pay = (hoursWorks*dollarsPerHour) + (overTime*1.5*dollarsPerHour)
        return(pay)
```

```
In [7]: payday(45,20)
```

```
Out[7]: 950.0
```

Exercise 4

Create Asserts to test your code from Exercise 2. Specifically, test the following test cases:

- Check whether the functions outputs 20 with inputs hoursWorks = 2 and dollarsPerHour = 10.
- Check whether the function does **not** output 410 if hoursWorks = 41 and dollarsPerHour = 10.

After all cases are passed successfully, print "Asserts Completed Successfully".

```
In [8]: retVal = payday(2, 10)
        assert(retVal == 20)

        retVal = payday(41, 10)
        assert(retVal != 410)

        print('Asserts completed successfully')
```

Asserts completed successfully

Exercise 5

Write code to count the number of times a "risk" word appears from the "risk_lexicon" in the string variable named "text." Please ignore case (i.e., you should lowercase everything). Lexicon-based analysis is popular because it quantifies various aspects of language.

The output of your code should look like the following:

Risk Count: 2

HINT: This exercise will require the use of a **for loop**, an **if statement**, and the string operations **.lower()** and **.split(" ")**.

```
In [9]: risk_lexicon = set(['danger', 'concern', 'risk', 'risky', 'doubt'])
        text = "CocaCola is a risky RISKY investment !!!!!"

        count = 0
        for item in risk_lexicon:
```

```
for word in text.split():  
    if item.lower() == word.lower(): count +=1  
print('Risk Cout:', count)
```

Risk Cout: 2