# 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 [107…

```
# Write code here
mbox = open("mbox.txt")

line_count = 0
unique_line_count = 0

for line in mbox:
    line_count += 1
    if "From:" in line:
        unique_line_count += 1

mbox.close()

print(f"The total number of lines in the file is: {line_count}")
print(f"The number of lines that contain the substring 'From' is: {unique_line_count}")
```

```
The total number of lines in the file is: 132044
The number of lines that contain the substring 'From' is: 1797
```

## Exercise 2

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

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

Next, append the string "SUM: k", wherekis 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 [114…
```
# If you mess up the file and need to try again, run this cell again to recreate the file
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 [115…
```
total = 0

num_file = open("numbers.txt", "r")

for line in num_file:
    number = int(line.strip())
    total += number

with open("numbers.txt", "a") as file:
    file.write(f"\nSum: {total}")

num_file.close()
```

Run the line below to check your work

In [116…
```
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 [27]:
```
# Test arguments
hours = 45 # You worked for 45 hours
pay = 10 # You make 10 dollars and hour
```

In [106…
```
# Write function here
def pay_calc(hoursWorks, dollarsPerHour):
    try:
        if hoursWorks <= 40:
            pay = hoursWorks * dollarsPerHour
            return pay
        elif hoursWorks > 40:
            pay = (hoursWorks - (hoursWorks - 40)) * dollarsPerHour
            pay_ot = (hoursWorks - 40) * (dollarsPerHour * 1.5)
            return pay + pay_ot
    except:
        return "You didn't enter a number!"
```

In [105…
```
# Call and print output of function here
overtime_pay = pay_calc(hours, pay)
regular_pay = pay_calc(20, 10)
incorrect_entry = pay_calc('apple', 'banana')

print(overtime_pay)
print(regular_pay)
print(incorrect_entry)
```

```
475.0
200
You didn't enter a number!
```

# Exercise 4

Create Asserts to test your code from Exercise 3. 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 [104…
```
# WRITE CODE HERE
assert(pay_calc(2, 10) == 20)
assert(pay_calc(41, 10) != 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 [103…
```
risk_lexicon = set(['danger', 'concern', 'risk', 'risky', 'doubt'])
text = "CocaCola is a risky RISKY investment !!!!"

counter = 0

for word in text.split(" "):
    if word.lower() in risk_lexicon:
        counter += 1

print(f"Risk Count: {counter}")
```

```
Risk Count: 2
```