

## LAB 04

### SUBMISSION INSTRUCTIONS

Submit 1 python file using the naming convention below (replace **JaneDoe** with your first and last name respectively):

- JaneDoe4.py

### QUESTION

1. Download **input.txt** and store it in the same location as the script file.
2. Write a program that will go through the lines in **input.txt**:
  - a. If a line begins with **From** (use regular expressions to find the lines)
    - i. Extract the email address, day, date, month, year, and time (please note that there should be a space after the word "**From** " to avoid logic errors).
    - ii. Send the details to a csv file named **output.csv** with the help of csv methods. This **output.csv** file should be stored in the same location as the script file.
    - iii. The contents of your **output.csv** file should be identical to **sample.csv** (include the header row).
  - b. If a line begins with **From:** (use regular expressions to find the lines)
    - i. Extract the email address (please note that the colon is part of the word because there are lines that begin with **From** without a colon at the end).
    - ii. Keep track of how many times an email is sent from a particular address with the help of a dictionary.
    - iii. Send your dictionary results to an output file called **output.txt**. This **output.txt** file should be stored in the same location as the script file.
    - iv. The contents of your **output.txt** file should be identical to **sample.txt** (include the last row showing total email count).

Please note:

- Write your program as a script (i.e., include the if `__name__ == '__main__':` block).
- Use the input and output file names provided in the instructions.
- Output files should be opened in write mode not append mode.
- Don't forget to close any files you open.
- Canvas shows the csv file as separated by tabs, but you should use the default ',' delimiter.
- Use `newline=''` to prevent blank rows from being inserted into your csv file (refer to the explanation of slide 17 of the lecture notes). It might appear to have no effect on your PC, but you should always write your scripts to accommodate issues it might encounter on other OS systems.
- I used 40 spaces for the email fields (e.g., `file.write(f'{"Email":40s} -Count \n')`) to achieve uniform formatting. This is preferable to using tab spaces as tab spaces might not be consistent across different systems (Refer to your week 2 string notes for details on this).