

## SI 206 Data-Oriented Programming

### Project Name: Working with File Data

#### Homework Objective:

To demonstrate the ability to:

- Manipulate files by reading and writing to them
- Organize data based on different keys
- Create histograms (counts) from clean data
- Sort data

#### Deliverables and Submission Process:

You will submit a single file called project1-206.py via Canvas. Late assignments are accepted, but will count towards your 6 day limit. You must also include a link to your Project 1 repository on GitHub.

#### Supporting Material: (See all material Google Drive - [Project1](#))

The [project1-206.py](#) file has the unittests already in place to develop and test your code.

Two data files [P1DataA.csv](#) and [P1DataB.csv](#) are provided as input to your code.

The file [ExampleFinalOutput.png](#) is also provided as an example of a successful run and [outfile.csv](#) is used as an example output file when the data is sorted by Last Name.

#### Background:

In this assignment you may use any Python constructs of your choosing. If you decided to use a Python collection that we have not covered in class, be prepared to explain your code to a instructor. (We can not help you debug random code from the Web.)

#### Steps:

**Fill in the code for different functions. Do not change any code in the main function.**

**You can run 206project1.py to see examples of output.**

1. `def getData(file):`  
    #Input: file name  
    #Output: return a list of dictionary objects where  
    #the keys are from the first row in the data. and the values are each of the other rows

#Note: The column headings will not change from the  
#test cases below, but the the data itself will  
#change (contents and size) in the different test  
#cases.

An example from a smaller data file would be:

```
{('First': 'Patience', 'Last': 'Hopkins', 'Email':  
'vehicula.Pellentesque.tincidunt@mollisneccursus.ca', 'Class': 'Senior', 'DOB': '7/1/26'),  
{('First': 'Fuller', 'Last': 'Harrell', 'Email': 'faucibus.ut@pharetra.ca', 'Class': 'Freshman',
```

```
'DOB': '3/15/36'}, {'First': 'Kelly', 'Last': 'Cole', 'Email': 'est.ac.mattis@Duis.org', 'Class': 'Junior', 'DOB': '7/28/14'}, {'First': 'Kylan', 'Last': 'Sparks', 'Email': 'eestas.ligula.Nullam@mus.edu', 'Class': 'Sophomore', 'DOB': '7/2/82'}]
```

2. `def mySort(data,col):`  
#Sort based on key/column  
#Input: list of dictionaries and col (key) to sort on  
#Output: Return the first item in the sorted list as a string of just: firstName lastName  
In an example from the list above with stored in the variable **data** the function call **mySort(data,'Last')** would return **Kelly Cole**. The same function call with **mySort(data,First)** would return **Fuller Harrell**.
3. `def classSizes(data):`  
# Create a histogram  
# Input: list of dictionaries  
# Output: Return a list of tuples sorted by the number of students in that class in  
# descending order  
# [('Senior', 26), ('Junior', 25), ('Freshman', 21), ('Sophomore', 18)]
4. `def findMonth(a):`  
# Find the most common birth month form this data  
# Input: list of dictionaries  
# Output: Return the month (1-12) that had the most births in the data
5. `def mySortPrint(a,col,fileName):`  
#Similar to mySort, but instead of returning single  
#Student, the sorted data is saved to a csv file.  
# as first,last,email  
#Input: list of dictionaries, col (key) to sort by and output file name  
#Output: No return value, but the file is written

### **1. Extra Credit – 10 points possible**

Find the average age (rounded) of the Students

```
# def findAge(a):  
# Input: list of dictionaries  
# Output: Return the average age of the students and round that age to the nearest  
# integer. You will need to work with the DOB to find the current age in years.
```

### **Integrity Policy:**

All materials submitted by students must be their own work - you may not submit material from previous semesters or examples taken from class or the Internet. Students may discuss the homework with others, but should not share code. If you work with others, make sure to indicate their name and the nature of the collaboration. Any instances of cheating will receive a 0 on the assignment and one letter grade deduction in the final course grade. If you are unsure

about the integrity of your submission, you have 48 hours after submission to withdraw your submission.