SI 206 Data-Oriented Programming

Project Name: Clean Data

Homework Objective:
To demonstrate the ability to:

- Manipulate files by reading and/or writing to them
- Organize data based on different keys
- Create histograms from clean data
- Find the average from multiple values
- Use String methods to sort data based on substrings

Deliverables and Submission Process:
You will submit a single file called 206project1.py via Canvas. Late assignments are accepted, but will count towards your 8 day limit. You must also include a link to your Project 1 repository on GitHub.

Supporting Material: (See all material Google Drive - Project1)
The 206project1.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 SampleScreenGrab.txt is also provided as an example of a successful run and outfile.csv is used as an example outfile 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:
You should 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
   #Ouput: return a list of dictionary objects where
   #the keys will come from the first row in the data.

   #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': 'egestas.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 the key to sort by.  Key will be “First”, “Last”, or “Email”
#Output: Return a string of the form 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 ordered by
# ClassName and Class size, e.g
# [('Senior', 26), ('Junior', 25), ('Freshman', 21), ('Sophomore', 18)]
```

4. def findDay(a):

```
# Find the most common day of the year to be born
# Input: list of dictionaries
# Output: Return the day of month (1-31) that is the
# most often seen in the DOB
```

5. 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 their current age.
```

6. def mySortPrint(a,col,fileName):

```
#Similar to mySort, but instead of returning single
#Student, all of the sorted data is saved to a csv file.
#Input: list of dictionaries, key to sort by and output file name
```

```
    #Output: Nothing to the screen, but it will create a new file called fileName
```

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.