Author: Martin Solum
Date: 06DEC2024Fri
Course: CIS129 Prog & Problem Solv I
Instructor: Brittany Griwzow
Assignment: Module 11 Lab
Exercise: 9.1, 9.2 and 9.3 in Deitel & Deitel

Table of Contents

Module 11 Lab Description

Instructions:

- Complete exercises 9.1, 9.2 and 9.3 in Deitel & Deitel.
- Submit completed Python code as a .py source file to D2L.
- Include this project within your GitHub account.

9.1 (Class Average: Writing Grades: Plain Text File)

Figure 3.2 presented a class-average script in which you could enter any number of grades followed by a sentinel value, then calculate the class average. Another approach would be to read the grades from a file. *In an IPython session, write code that enables you to store any number of grades into a grades.txt plain text file.

9.2 (Class Average: Reading Grades: Plain Text File)

In an *IPython session, write code that reads the grades from the grades.txt file* you created in the previous exercise. Display the individual grades and their total, count and average.

9.3 (Class Average: Writing Student Records: CSV File)

An instructor teaches a class in which each student takes three exams. The instructor would like to store this information in a file named grades.csv for later use. *Write code that enables an instructor to enter each student's first name and last name as strings and the student's three exam grades as integers.* Use the csv module to write each record into the grades.csv file. Each record should be a single line of text in the following CSV format:

firstname,lastname,exam1grade,exam2grade,exam3grade

Module 11 Lab WriteUp

The assignment has 3 major parts…

- 9.1 Class Average: Writing Grades: Plain Text File

```
with open('grades.txt', mode='w') as students:
    students.write('Burnham Michael 91 92 93\n')
    students.write('Carter Samantha 95 97 96\n')
    students.write('Conner Sarah 77 78 79\n')
    students.write('Drummer Camina 89 87 88\n')
    students.write('Emmagan Teyla 83 85 84\n')
    students.write('Valentine Beka 91 94 92\n')
    students.write('Weir Elizabeth 95 96 94\n')
```

- 9.2 Class Average: Reading Grades: Plain Text File

```
with open('grades.txt', mode='r') as students:
    labels = f'{"Last name":<15}{"First Name":<15}{"Score #1":^12}
{"Score #2":^12}'
    labels = labels + f'{"Score #3":^12}{"total":^12}{"count":^12}
{"average":^12}'
    print( labels )
    for record in students:
        lastName, firstName, score1, score2, score3 = record.split()
        scores = pd.Series( [int(score1), int(score2),
int(score3)] )
        total = scores.sum()
        count = scores.count()
        average = scores.mean()
        str = f'{lastName:<15}{firstName:<15}{score1:^12}
{score2:^12}{score3:^12}'
        str = str + f'{total:^12}{count:^12}{average:^12,.2f}'
        print( str )
```

9.3 Class Average: Writing Student Records: CSV File

```
with open( 'grades.csv', mode='w', newline='' ) as students:
 writer = csv.writer( students )
 writer.writerow( ["Burnham", "Michaels", 91, 92, 93] )
 writer.writerow( ["Carter", "Samantha", 95, 97, 96] )
 writer.writerow( ["Conner", "Sarah", 77, 78, 79] )
 writer.writerow( ["Drummer", "Camina", 89, 87, 88] )
 writer.writerow( ["Emmagan", "Teyla", 83, 85, 84] )
 writer.writerow( ["Valentine", "Beka", 91, 94, 92] )
 writer.writerow( ["Weir", "Elizabeth", 95, 96, 94] )
```

The I wrote a complete program to illustrate use of the code specified above.
Basically, I wrote a program that handles 2 argument arrays { for now }.

If the program is called with a -r <filename> argument array the program:
    * reads the file.
    * calculate the class average for each of the students.
    * reports the class average for each student and the respective letter grades.

if the program is called with the -w <filename> argument array the program:
    * solicits data from the user regarding the students, specifically:
        * student’s first name (string)
        * student's last name (string)
        * student's 3 exam scores (int)
        * calculate the class average for each of the students.
        * reports the class average for each student and the respective letter grades.
    * writes the solicited data to the given data file.

if the program is called with a -a <filename> argument array the program:
    * reads the file.
    * solicits data from the user regarding the students, specifically:
        * student’s first name (string)
        * student's last name (string)
        * student's 3 exam scores (int)
        * calculate the class average for each of the students.
        * reports the class average for each student and the respective letter grades.
 * rewrites the data file, appending to data file the new information provided by the user.

NOTE: A caveat about txt vs csv files...
* The system can read & write simple text data files.
* The system can read & write simple csv data files.
* The system reads & writes files with the following assumptions:
        * .txt file suffix denotes a simple text data file.
        * .csv file suffix denotes a comma separated values formatted file.
        * The system assumes the suffix is correct and acts accordingly.

Further notes on this assignment.

This assignment was difficult because of environment errors. Basically the latest pandas library is incompatible with the latest numpy library. I found this combination of systems versions worked best:

Python 3.12.3
Pandas 2.2.2
Numpy 1.26.4

If I had more time I would refactor the read methods to avoid repeating code unnecessarily.