

## PROJECT INTRODUCTION

This is a Python completion project at One Campus Academy, which is intended to test my python fundamentals knowledge

## PROJECT SCENARIO

Jerome is a teacher at the Grand Rapids High School. Due to the lack of teachers willing to accept in-person teaching positions during the Covid era, Jerome has had to teach Math, Chemistry, Biology and Physics for the 8th grade students. The school term is made up of 96 class sessions and attendance is taken each day. Students earn 1 point for each class attendance. Jerome computes percentage class attendance and awards proportional score with a maximum possible score of 12 points for the class attendance contribution to total performance for the school term. There are class quizzes every week (19 in total for the term). Each quiz is graded over 10 maximum points. And a max aggregate score of 30 for the quiz contribution to total performance for the school term Homework is assigned every week (19 in total for the term). Each homework is graded over 10 maximum points. And a max aggregate score of 15 points for the homework contribution to total performance for the school term

## SCRIPT UTILITY

- Get the full performance records of any student by simply providing the student's name or ID
- Get any student's performance for any given subject by simply providing student name/ID and subject name
- Get the list of students in any of the grade categories (A, B, C, D, E, F)
- Get the list of students Statuses (Pass, Fail, Retake)
- Obtain a response from the script for any student to determine if they passed, failed, or need to retake the subject

## SCRIPT CAPABILITIES:

- enables the teacher to manually load each students scores by subject for quiz, homework, attendance, and exam using the python input() function. Use as data, the records in the Student Scores table below (Table1.0)
- Automatically computes the Average Score, GPA (0 to 5.0), Grade and Status (Pass, Fail, Retake) and stores in a container for each student as per the provided rating scales below
- Holds each student's performance records in a container
- Holds each subject performance records in a container for all students
- Holds ALL student performance records for ALL subjects in a container. Note: this container should hold all student scores for subjects quiz, homework and exam scores, Average Score, GPA, Grade and Status. All in a Container of containers

NOTE:Jerome would also like to have the functionality to query the containers above to get any specific performance records.

## Below are the students name and ID

1. Bett James GR-0483
2. Namukolo Abrams GR-0484
3. Vera Abutu GR-0485
4. Kwame Doga GR-0486
5. Lukeman Ahmad GR-0487
6. Akin Torey GR-0488
7. Luke Brant GR-0489
8. James Kenyata GR-0490
9. Ngugi Tionga GR-0491
10. Okoro Eze GR-0492
11. Agatha Chiluba GR-0493
12. Mangu Joseph GR-0494
13. Longe Jethro GR-0495
14. Florence Giwa GR-0496
15. Vetiva Lucent GR-0497
16. Melody Braimoh GR-0498
17. Victor Ihab GR-0499
18. Mimi Trucker GR-0500
19. Maguel Peter GR-0501
20. Wellington Zuba GR-0502

In [1]:

```
# importing a package to compute the overall mean of the students performance
import statistics

# Creating empty dictionaries to hold the overall performance and the scores for each student
perfList = []
studSubsDict = dict()

# defining a function for the student's name, ID, subject, assessment type and their scores
def scoreAggregator(name, ID, subject, assessTypeList, scoreList):

    tempDict = dict()
    sList = scoreList.split(',')
    sList = [float(x) for x in sList]

    resDict = dict(zip(assessTypeList, sList))
    tempDict[subject] = resDict

    return tempDict

# defining a function to compute the average score for each student
def avgScoreComput(studname, sDictKeys):
    studName = studname
    qContrib = ''
    hContrib = ''
    aContrib = ''
    eContrib = ''
    ScoreList = []
    grade = ''
```

```

gpa = ''
status = ''

for rec in list(sDictKeys):
    subject = list(rec.keys())[0]
    subjScores = list(rec.values())[0]

    # Creating a for Loop to get the average score of each performance type
    for sub in list(subjScores.keys()):
        if sub=='Quiz':
            qContrib = (float(subjScores[sub])/190)*30
        elif sub == 'HW':
            hContrib = (float(subjScores[sub])/190)*15

        elif sub == 'ATTND':
            aContrib = (float(subjScores[sub])/96)*12
        elif sub == 'Exam':
            eContrib = (float(subjScores[sub])/100)*43
        else:
            print('assessment type unkown')
    avgScore = qContrib +hContrib +aContrib + eContrib
    ScoreList.append(avgScore)
    print(avgScore)
    Score = statistics.mean(ScoreList)
    Score = round(Score, 2)

print(f'this is the mean score of the student: {Score}')
# Getting the gpa of each student
gpa = (Score/100)*5
if Score >= 70:
    grade = 'A'
elif Score >= 69:
    grade = 'B'
elif Score >= 59:
    grade = 'C'
elif Score >= 49:
    grade = 'D'
else:
    grade = 'F'

if grade in ['A','B','C','D']:
    status = 'Pass'
else:
    status = 'Fail'
return ScoreList, Score, grade, gpa, status

```

In [101...

```

# writing codes to input each student details

name = input('student name')
ID = str(input('student ID'))
subject = input('subject')
assessTypeList = ['Quiz', 'HW', 'ATTND', 'Exam'] #input('assessment type List')
scoresList = input('enter scores in the following order; Quiz, HW, ATTND, Exam')
flag = int(input('set flag'))
# If flag is set to 1, code aggregates subject wise score. If flag is set to 2 it c

if flag == 1:
    tempDict = scoreAggregator(name, ID, subject, assessTypeList, scoresList)
    perfList.append(tempDict)
else:
    name_ID = name + '_' + str(ID)
    res = avgScoreComput(name_ID, perfList)

```

```

print(res)
scorecard = dict()
ScoreList, Score, grade, gpa, status = res
scorecard['Subject Scores'] = ScoreList
scorecard['Overall Score'] = Score
scorecard['Grade'] = grade
scorecard['GPA'] = gpa

scorecard['Status'] = status

perfList.append(scorecard)

studSubsDict[name_ID]=perfList
perfList = []

print(perfList)
print(studSubsDict)

```

student nameWellington Zuba  
student IDGR-0502  
subject  
enter scores in the following order; Quiz, HW, ATTND, Exam  
set flag2  
89.50631578947369  
83.22421052631579  
96.20157894736843  
85.98973684210526  
this is the mean score of the student: 88.73  
([89.50631578947369, 83.22421052631579, 96.20157894736843, 85.98973684210526], 88.73, 'A', 4.4365000000000006, 'Pass')  
[]  
{'Bett James\_GR-0483': [{'Math': {'Quiz': 123.0, 'HW': 178.0, 'ATTND': 87.0, 'Exam': 67.0}}, {'Chemistry': {'Quiz': 133.0, 'HW': 145.0, 'ATTND': 87.0, 'Exam': 98.0}}, {'Biology': {'Quiz': 134.0, 'HW': 124.0, 'ATTND': 89.0, 'Exam': 76.0}}, {'Physics': {'Quiz': 134.0, 'HW': 156.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Subject Scores': [73.15868421052632, 85.46236842105263, 74.75236842105264, 77.58368421052631], 'Overall Score': 77.74, 'Grade': 'A', 'GPA': 3.887, 'Status': 'Pass'}], 'Namukolo Abrams\_GR-0484': [{'Math': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 167.0, 'ATTND': 76.0, 'Exam': 89.0}}, {'Biology': {'Quiz': 143.0, 'HW': 166.0, 'ATTND': 67.0, 'Exam': 78.0}}, {'Physics': {'Quiz': 145.0, 'HW': 177.0, 'ATTND': 77.0, 'Exam': 78.0}}, {'Subject Scores': [80.18894736842105, 82.1121052631579, 77.59921052631579, 80.03342105263158], 'Overall Score': 79.98, 'Grade': 'A', 'GPA': 3.9990000000000006, 'Status': 'Pass'}], 'Vera Abutu\_GR-0485': [{'Math': {'Quiz': 123.0, 'HW': 155.0, 'ATTND': 76.0, 'Exam': 78.0}}, {'Chemistry': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 87.0, 'Exam': 99.0}}, {'Biology': {'Quiz': 134.0, 'HW': 176.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Physics': {'Quiz': 134.0, 'HW': 177.0, 'ATTND': 88.0, 'Exam': 76.0}}, {'Subject Scores': [74.6978947368421, 89.52394736842106, 80.02263157894737, 78.81157894736842], 'Overall Score': 80.76, 'Grade': 'A', 'GPA': 4.038, 'Status': 'Pass'}], 'Kwame Doga\_GR-0486': [{'Math': {'Quiz': 132.0, 'HW': 123.0, 'ATTND': 87.0, 'Exam': 69.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 156.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Biology': {'Quiz': 123.0, 'HW': 177.0, 'ATTND': 89.0, 'Exam': 87.0}}, {'Physics': {'Quiz': 134.0, 'HW': 155.0, 'ATTND': 76.0, 'Exam': 89.0}}, {'Subject Scores': [71.09763157894737, 77.58368421052631, 81.92973684210526, 81.16473684210527], 'Overall Score': 77.94, 'Grade': 'A', 'GPA': 3.897, 'Status': 'Pass'}], 'Lukeman Ahmad\_GR-0487': [{'Math': {'Quiz': 155.0, 'HW': 166.0, 'ATTND': 77.0, 'Exam': 99.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 177.0, 'ATTND': 89.0, 'Exam': 90.0}}, {'Biology': {'Quiz': 134.0, 'HW': 156.0, 'ATTND': 89.0, 'Exam': 90.0}}, {'Physics': {'Quiz': 170.0, 'HW': 180.0, 'ATTND': 80.0, 'Exam': 95.0}}, {'Subject Scores': [89.77394736842106, 84.95657894736843, 83.29868421052632, 91.90263157894736], 'Overall Score': 87.48, 'Grade': 'A', 'GPA': 4.3740000000000006, 'Status': 'Pass'}], 'Akin Torey\_GR-0488': [{'Math': {'Quiz': 132.0, 'HW': 167.0, 'ATTND': 89.0, 'Exam': 78.0}}, {'Chemistry': {'Quiz': 144.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Biology': {'Quiz': 155.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Subject Scores': [78.69131578947369, 80.03105263157894, 82.62789473684211], 'Overall Score': 8

0.45, 'Grade': 'A', 'GPA': 4.0225, 'Status': 'Pass']], 'Luke Brant\_GR-0489': [{'Math': {'Quiz': 188.0, 'HW': 178.0, 'ATTND': 89.0, 'Exam': 78.0}}, {'Chemistry': {'Quiz': 144.0, 'HW': 187.0, 'ATTND': 90.0, 'Exam': 87.0}}, {'Biology': {'Quiz': 145.0, 'HW': 189.0, 'ATTND': 77.0, 'Exam': 87.0}}, {'Physics': {'Quiz': 176.0, 'HW': 177.0, 'ATTND': 89.0, 'Exam': 90.0}}, {'Subject Scores': [88.40184210526316, 86.16, 84.85078947368422, 91.58815789473684], 'Overall Score': 87.75, 'Grade': 'A', 'GPA': 4.387499999999999, 'Status': 'Pass'}], 'James Kenyatta\_GR-0490': [{'Math': {'Quiz': 156.0, 'HW': 170.0, 'ATTND': 89.0, 'Exam': 78.0}}, {'Chemistry': {'Quiz': 178.0, 'HW': 188.0, 'ATTND': 76.0, 'Exam': 67.0}}, {'Biology': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 94.0, 'Exam': 93.0}}, {'Physics': {'Quiz': 176.0, 'HW': 188.0, 'ATTND': 76.0, 'Exam': 76.0}}, {'Subject Scores': [82.71763157894736, 81.25736842105263, 90.42421052631579, 84.81157894736842], 'Overall Score': 84.8, 'Grade': 'A', 'GPA': 4.24, 'Status': 'Pass'}], 'Ngugi Tionga\_GR-0491': [{'Math': {'Quiz': 155.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 123.0, 'ATTND': 77.0, 'Exam': 76.0}}, {'Biology': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Physics': {'Quiz': 155.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Subject Scores': [81.7678947368421, 73.17342105263157, 81.04894736842105, 82.62789473684211], 'Overall Score': 79.65, 'Grade': 'A', 'GPA': 3.9825000000000004, 'Status': 'Pass'}], 'Okoro Eze\_GR-0492': [{'Math': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 91.0, 'Exam': 92.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 155.0, 'ATTND': 77.0, 'Exam': 89.0}}, {'Biology': {'Quiz': 134.0, 'HW': 177.9, 'ATTND': 90.0}}, {'Physics': {'Quiz': 134.0, 'HW': 121.0, 'ATTND': 76.0, 'Exam': 95.0}}, {'Subject Scores': [89.61921052631578, 81.28973684210527, 84.72263157894736, 81.06052631578947], 'Overall Score': 84.17, 'Grade': 'A', 'GPA': 4.2085, 'Status': 'Pass'}], 'Agatha Chiluba\_GR-0493': [{'Math': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 67.0, 'Exam': 98.0}}, {'Chemistry': {'Quiz': 123.0, 'HW': 156.0, 'ATTND': 87.0, 'Exam': 88.0}}, {'Biology': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Physics': {'Quiz': 167.0, 'HW': 176.0, 'ATTND': 78.0, 'Exam': 78.0}}, {'Subject Scores': [89.1992105263158, 80.45184210526315, 81.04894736842105, 83.55315789473684], 'Overall Score': 83.56, 'Grade': 'A', 'GPA': 4.178, 'Status': 'Pass'}], 'Mangu Joseph\_GR-0494': [{'Math': {'Quiz': 123.0, 'HW': 167.0, 'ATTND': 78.0, 'Exam': 88.0}}, {'Chemistry': {'Quiz': 155.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Biology': {'Quiz': 165.0, 'HW': 176.0, 'ATTND': 88.0, 'Exam': 97.0}}, {'Physics': {'Quiz': 145.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Subject Scores': [80.19526315789474, 82.62789473684211, 92.65736842105264, 81.05736842105263], 'Overall Score': 84.13, 'Grade': 'A', 'GPA': 4.2065, 'Status': 'Pass'}], 'Longe Jethro\_GR-0495': [{'Math': {'Quiz': 145.0, 'HW': 168.0, 'ATTND': 88.0, 'Exam': 76.0}}, {'Chemistry': {'Quiz': 124.0, 'HW': 156.0, 'ATTND': 76.0, 'Exam': 89.0}}, {'Biology': {'Quiz': 134.0, 'HW': 156.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Physics': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Subject Scores': [79.83789473684212, 79.66473684210527, 77.58368421052631, 81.04894736842105], 'Overall Score': 79.53, 'Grade': 'A', 'GPA': 3.9765, 'Status': 'Pass'}], 'Florence Giwa\_GR-0496': [{'Math': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 90.0}}, {'Chemistry': {'Quiz': 134.0, 'HW': 177.0, 'ATTND': 88.0, 'Exam': 87.0}}, {'Biology': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 77.0}}, {'Physics': {'Quiz': 167.0, 'HW': 188.0, 'ATTND': 93.0, 'Exam': 79.0}}, {'Subject Scores': [85.77894736842106, 83.54157894736841, 82.7942105263158, 86.80552631578948], 'Overall Score': 84.73, 'Grade': 'A', 'GPA': 4.2365, 'Status': 'Pass'}], 'Vetiva Lucent\_GR-0497': [{'Math': {'Quiz': 178.0, 'HW': 177.0, 'ATTND': 77.0, 'Exam': 69.0}}, {'Chemistry': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Biology': {'Quiz': 178.0, 'HW': 177.0, 'ATTND': 77.0, 'Exam': 69.0}}, {'Physics': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 76.0}}, {'Subject Scores': [81.37394736842106, 83.65421052631578, 81.37394736842106, 82.36421052631579], 'Overall Score': 82.19, 'Grade': 'A', 'GPA': 4.1095, 'Status': 'Pass'}], 'Melody Braimoh\_GR-0498': [{'Math': {'Quiz': 167.0, 'HW': 178.0, 'ATTND': 66.0, 'Exam': 87.0}}, {'Chemistry': {'Quiz': 145.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 87.0}}, {'Biology': {'Quiz': 156.0, 'HW': 176.0, 'ATTND': 88.0, 'Exam': 90.0}}, {'Physics': {'Quiz': 167.0, 'HW': 177.0, 'ATTND': 89.0, 'Exam': 76.0}}, {'Subject Scores': [86.08105263157894, 84.48894736842105, 88.22631578947369, 84.1471052631579], 'Overall Score': 85.74, 'Grade': 'A', 'GPA': 4.287, 'Status': 'Pass'}], 'Victor Ihab\_GR-0499': [{'Math': {'Quiz': 156.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Chemistry': {'Quiz': 176.0, 'HW': 167.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Biology': {'Quiz': 167.0, 'HW': 178.0, 'ATTND': 76.0, 'Exam': 87.0}}, {'Physics': {'Quiz': 167.0, 'HW': 155.0, 'ATTND': 89.0, 'Exam': 77.0}}, {'Subject Scores': [83.65421052631578, 85.94368421052631, 87.33105263157894, 82.84026315789474], 'Overall Score': 84.94, 'Grade': 'A', 'GPA': 4.247, 'Status': 'Pass'}], 'Mimi Trucke

```
r_GR-0500': [{'Math': {'Quiz': 167.0, 'HW': 165.0, 'ATTND': 77.0, 'Exam': 86.0}},
{'Chemistry': {'Quiz': 157.0, 'HW': 178.0, 'ATTND': 89.0, 'Exam': 79.0}}, {'Biology': {'Quiz': 178.0, 'HW': 177.0, 'ATTND': 78.0, 'Exam': 89.0}}, {'Physics': {'Quiz': 178.0, 'HW': 187.0, 'ATTND': 92.0, 'Exam': 90.0}}, {'Subject Scores': [85.99973684210525, 83.93710526315789, 90.09894736842105, 93.06842105263158], 'Overall Score': 88.28, 'Grade': 'A', 'GPA': 4.414, 'Status': 'Pass'}], 'Maguel Peter_GR-0501': [{'Math': {'Quiz': 180.0, 'HW': 178.0, 'ATTND': 88.0, 'Exam': 79.0}}, {'Chemistry': {'Quiz': 167.0, 'HW': 178.0, 'ATTND': 89.0, 'Exam': 89.0}}, {'Chemistry': {'Quiz': 178.0, 'HW': 189.0, 'ATTND': 67.0, 'Exam': 89.0}}, {'Biology': {'Quiz': 178.0, 'HW': 189.0, 'ATTND': 89.0, 'Exam': 90.0}}, {'Subject Scores': [87.44368421052631, 89.81605263157894, 89.67131578947368, 92.85131578947369], 'Overall Score': 89.95, 'Grade': 'A', 'GPA': 4.4975000000000005, 'Status': 'Pass'}], 'Wellington Zuba_GR-0502': [{'Math': {'Quiz': 189.0, 'HW': 167.0, 'ATTND': 76.0, 'Exam': 86.0}}, {'Chemistry': {'Quiz': 167.0, 'HW': 156.0, 'ATTND': 88.0, 'Exam': 78.0}}, {'Biology': {'Quiz': 176.0, 'HW': 188.0, 'ATTND': 88.0, 'Exam': 99.0}}, {'Physics': {'Quiz': 176.0, 'HW': 166.0, 'ATTND': 89.0, 'Exam': 79.0}}, {'Subject Scores': [89.50631578947369, 83.22421052631579, 96.20157894736843, 85.98973684210526], 'Overall Score': 88.73, 'Grade': 'A', 'GPA': 4.4365000000000006, 'Status': 'Pass'}]]
```

```
In [ ]: # res = avgScoreComput('Bola_G5667', studSubsDict['Bola_G5667'])
```

```
In [105... # Testing if this would work: Get the full performance records of any student by sim
studSubsDict['Bett James_GR-0483']
```

```
Out[105... [{'Math': {'Quiz': 123.0, 'HW': 178.0, 'ATTND': 87.0, 'Exam': 67.0}},
{'Chemistry': {'Quiz': 133.0, 'HW': 145.0, 'ATTND': 87.0, 'Exam': 98.0}},
{'Biology': {'Quiz': 134.0, 'HW': 124.0, 'ATTND': 89.0, 'Exam': 76.0}},
{'Physics': {'Quiz': 134.0, 'HW': 156.0, 'ATTND': 88.0, 'Exam': 77.0}},
{'Subject Scores': [73.15868421052632,
85.46236842105263,
74.75236842105264,
77.58368421052631],
'Overall Score': 77.74,
'Grade': 'A',
'GPA': 3.887,
'Status': 'Pass'}]
```

```
In [106... # Testing if this would work: Get any student's performance for any given subject by
# Getting the score Bett James Score for chemistry
studSubsDict['Bett James_GR-0483'][0].values()
```

```
Out[106... dict_values(['Quiz': 123.0, 'HW': 178.0, 'ATTND': 87.0, 'Exam': 67.0])
```

```
In [122... # Get the List of students in any of the grade categories (A, B, C, D, E, F)
#studSubsDict[Grade = A]
for i in studSubsDict:
    if i[grade] == 'A':
        print(i)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13512\3971764211.py in <module>
      2 #studSubsDict[Grade = A]
      3 for i in studSubsDict:
----> 4     if i[grade] == 'A':
      5         print(i)
```

**TypeError:** string indices must be integers

```
In [123... #Get the list of students Statuses (Pass, Fail, Retake)
for i in studSubsDict:
    if i[status] == 'Pass':
        print(i)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13512\1953753006.py in <module>
      1 #Get the list of students Statuses (Pass, Fail, Retake)
      2 for i in studSubsDict:
----> 3     if i[status] == 'Pass':
      4         print(i)
```

**TypeError:** string indices must be integers

```
In [114... #Obtain a response from the script for any student to determine if they passed, fail
list(studSubsDict['Bett James_GR-0483'])[4].values())[-1]
```

Out[114... 'Pass'

In [ ]: