

STUDENT PERFORMANCE DATA ANALYSIS

Part 1: Raw Logic

```
In [45]: # Initialize an empty dictionary to hold student data
student_data = {}

# Take the number of students in class, subjects offered and subject label
no_of_studs = int(input('How many students will be in this record: '))
no_of_subjects = int(input('How many subjects will be in this record: '))
sub_label = [input('Enter name for subject {}: '.format(i+1)).strip().title() for i in range(no_of_subjects)]

# Collect individual student data
for i in range(no_of_studs):
    student_record = {}
    student_record['name'] = input('What is the name of this student: ')
    for subject in sub_label:
        student_record['{}'.format(subject)] = int(input('What was the score in {}: '.format(subject)))
    print('record taken successfully {}'.format(student_record))

# Add this record to the student data list
student_data.append(student_record)
print(student_data)
```

```
record taken successfully {'name': 'James', 'Maths': 78, 'English': 85, 'Physics': 68}
record taken successfully {'name': 'Faith', 'Maths': 85, 'English': 56, 'Physics': 54}
record taken successfully {'name': 'Stanley', 'Maths': 63, 'English': 54, 'Physics': 75}
record taken successfully {'name': 'Busayo', 'Maths': 78, 'English': 86, 'Physics': 56}
record taken successfully {'name': 'Stephanie', 'Maths': 78, 'English': 56, 'Physics': 68}
[{'name': 'James', 'Maths': 78, 'English': 85, 'Physics': 68}, {'name': 'Faith', 'Maths': 85, 'English': 56, 'Physics': 54}, {'name': 'Stanley', 'Maths': 63, 'English': 54, 'Physics': 75}, {'name': 'Busayo', 'Maths': 78, 'English': 86, 'Physics': 56}, {'name': 'Stephanie', 'Maths': 78, 'English': 56, 'Physics': 68}]
```

```
In [51]: # Calculation for Each Student's Average
for student in student_data:
    scores = [score for subject, score in student.items() if subject != 'name']
    average = sum(scores) / len(scores)
    student['average'] = round(average, 2)

student_data
```

```
Out[51]: [{ 'name': 'James',
            'Maths': 78,
            'English': 85,
            'Physics': 68,
            'average_score': 77.0,
            'average': 77.0},
          { 'name': 'Faith',
            'Maths': 85,
            'English': 56,
            'Physics': 54,
            'average_score': 65.0,
            'average': 65.0},
          { 'name': 'Stanley',
            'Maths': 63,
            'English': 54,
            'Physics': 75,
            'average_score': 64.0,
            'average': 64.0},
          { 'name': 'Busayo',
            'Maths': 78,
            'English': 86,
            'Physics': 56,
            'average_score': 73.33,
            'average': 73.33},
          { 'name': 'Stephanie',
            'Maths': 78,
            'English': 56,
            'Physics': 68,
            'average_score': 67.33,
            'average': 67.33}]
```

```
In [53]: # Assign grades to each student
for record in student_data:
    if record['average'] >=80 and record['average'] <=100:
        grade = "A - EXCELLENT"
    elif record['average'] >=70 and record['average'] <=79:
        grade = "B - VERY GOOD"
    elif record['average'] >=60 and record['average'] <=69:
        grade = "C - GOOD"
    elif record['average'] >=50 and record['average'] <=59:
        grade = "D - FAIR - Needs Improvement"
    else:
        grade = "F - POOR - Work Harder"
    record['grade'] = grade
```

```
student_data
```

```
Out[53]: [{ 'name': 'James',
            'Maths': 78,
            'English': 85,
            'Physics': 68,
            'average_score': 77.0,
            'average': 77.0,
            'grade': 'B - VERY GOOD'},
          { 'name': 'Faith',
            'Maths': 85,
            'English': 56,
            'Physics': 54,
            'average_score': 65.0,
            'average': 65.0,
            'grade': 'C - GOOD'},
          { 'name': 'Stanley',
            'Maths': 63,
            'English': 54,
            'Physics': 75,
            'average_score': 64.0,
            'average': 64.0,
            'grade': 'C - GOOD'},
          { 'name': 'Busayo',
            'Maths': 78,
            'English': 86,
            'Physics': 56,
            'average_score': 73.33,
            'average': 73.33,
            'grade': 'B - VERY GOOD'},
          { 'name': 'Stephanie',
            'Maths': 78,
            'English': 56,
            'Physics': 68,
            'average_score': 67.33,
            'average': 67.33,
            'grade': 'C - GOOD'}]
```

```
In [55]: # Summary for Each student
for record in student_data:
    print("Hi {}, you scored an average of {}, your grade is {}".format(record['name'], record['average'], record['grade']))
```

Hi James, you scored an average of 77.0, your grade is B - VERY GOOD
Hi Faith, you scored an average of 65.0, your grade is C - GOOD
Hi Stanley, you scored an average of 64.0, your grade is C - GOOD
Hi Busayo, you scored an average of 73.33, your grade is B - VERY GOOD
Hi Stephanie, you scored an average of 67.33, your grade is C - GOOD

PART 2: Refactor Using Functions

```
In [107... # Define a function that collects student data: collect_record
def collect_record():
    ''' Functions returns a list of dictionaries with each dictionary as a student record with key-value pairs of their details.
    Takes one argument which is the number of students in the class'''

    # Collect individual student data as records - first initiate an empty list
    student_data_2 = []
    no_of_studs = int(input('How many students will be in this record: '))
    no_of_subjects = int(input('How many subjects will be in this record: '))
    sub_label = [input('Enter name for subject {}: '.format(i+1)).strip().title() for i in range(no_of_subjects)]
    print('collected necessary data for record creation... \n'.capitalize())

    # Loop through the number of students to assign record for each
    for i in range(no_of_studs):
        student_record = {}
        print('Now, provide student record details according to the prompt')
        student_record['name'.title()] = input('What is the name of student {}: '.format(i+1))
        for subject in sub_label:
            student_record['{}_score'.format(subject)] = int(input('What was the score in {}: '.format(subject)))

        # Add student record to the list
        student_data_2.append(student_record)
        print('record taken successfully {} \n'.format(student_record))

    # return the student data
    return student_data_2
```

```
In [119... # Define a function that calculates the average score of each student
def calculate_average(scores):
    """This returns the average score from list (dataset) of records collected from the student's exam scores.
    It takes one argument which is the score of the student"""

    # Calculate the average from various subjects
    average = sum(scores) / len(scores)
    return average
```

In [121...

```
# Define a function that the grade of each student based on their average score
def assign_grade(avg):
    """ This returns the student's grade from the exam.
    It takes one argument which is the column of average score of the students"""

    # Logical statement to determine grade
    if avg >=80 and avg <=100:
        grade = "A"
    elif avg >=70 and avg <=79:
        grade = "B"
    elif avg >=60 and avg <=69:
        grade = "C"
    elif avg >=50 and avg <=59:
        grade = "D"
    else:
        grade = "F"
    return grade
```

In [123...

```
# Define a function that generates remark for each student based on their performance
def generate_remark(grade):
    ''' This returns a comment of the student/'s performance in the exam.
    It takes one argument which is the grade of the student'''

    # Logical statement to determine grade
    if grade == 'A':
        remark = "EXCELLENT"
    elif grade == 'B':
        remark = "VERY GOOD"
    elif grade == 'C':
        remark = "GOOD"
    elif grade == 'D':
        remark = "FAIR - Needs Improvement"
    else:
        remark = "POOR - Work Harder"
    return remark
```

In [125...

```
# Define a function that gives a summary table of student performance
def summarize_students(data):
    """This returns a summary table of the overall student's performance in the exam.
    It takes one argument which is the dataset of the students in the class"""

    # import necessary libraries
    import pandas as pd
```

```
# Create a pandas Dataframe of the data
summary_table = pd.DataFrame(data)

return summary_table
```

```
In [139... # Let us now try out these functions - using the collect record function and passing the number of student in class
exam_record = collect_record()

# Add a new record of the average score for each student
for student in exam_record:
    # Generate a list of scores for each student
    scores = [score for subject, score in student.items() if subject != 'Name']
    student['average_score'.title()] = round(calculate_average(scores), 1)
print("This is the exam record with student's average score {} \n".format(exam_record))

# Now, Let us assign them grades based on their average scores
print('ASSIGNING GRADES...')
for student in exam_record:
    student['grade'.title()] = assign_grade(student['average_score'.title()])

exam_record

# Let us generate remark for each student
for student in exam_record:
    student['remark'.title()] = generate_remark(student['grade'.title()])
    print('{} record shows performance as {} \n'.format(student['name'.title()], student['remark'.title()]))

exam_record

# Now, Let us conclude part 2 by viewing a summary table of the student data
stud_summ_table = summarize_students(exam_record)

print(stud_summ_table)
print('the above is the exam record of the selected student and the grades'.title())
```

Collected necessary data for record creation...

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Joanna', 'Maths_score': 75, 'English_score': 68, 'Science_score': 92, 'Arts_score': 72, 'Economics_score': 85}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Justus', 'Maths_score': 85, 'English_score': 78, 'Science_score': 89, 'Arts_score': 72, 'Economics_score': 86}

Now, provide student record details according to the prompt

record taken successfully {'Name': 'Julius', 'Maths_score': 35, 'English_score': 55, 'Science_score': 40, 'Arts_score': 32, 'Economics_score': 25}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Jamez', 'Maths_score': 50, 'English_score': 50, 'Science_score': 53, 'Arts_score': 58, 'Economics_score': 60}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Joyce', 'Maths_score': 62, 'English_score': 68, 'Science_score': 58, 'Arts_score': 72, 'Economics_score': 60}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Janetti', 'Maths_score': 75, 'English_score': 73, 'Science_score': 78, 'Arts_score': 74, 'Economics_score': 80}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Joshua', 'Maths_score': 85, 'English_score': 45, 'Science_score': 56, 'Arts_score': 68, 'Economics_score': 51}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Joseph', 'Maths_score': 42, 'English_score': 25, 'Science_score': 35, 'Arts_score': 61, 'Economics_score': 20}

Now, provide student record details according to the prompt
record taken successfully {'Name': 'Jennifer', 'Maths_score': 63, 'English_score': 85, 'Science_score': 65, 'Arts_score': 82, 'Economics_score': 72}

Now, provide student record details according to the prompt

record taken successfully {'Name': 'Johnny', 'Maths_score': 62, 'English_score': 65, 'Science_score': 68, 'Arts_score': 64, 'Economics_score': 50}

This is the exam record with student's average score [{'Name': 'Joanna', 'Maths_score': 75, 'English_score': 68, 'Science_score': 92, 'Arts_score': 72, 'Economics_score': 85, 'Average_Score': 78.4}, {'Name': 'Justus', 'Maths_score': 85, 'English_score': 78, 'Science_score': 89, 'Arts_score': 72, 'Economics_score': 86, 'Average_Score': 82.0}, {'Name': 'Julius', 'Maths_score': 35, 'English_score': 55, 'Science_score': 40, 'Arts_score': 32, 'Economics_score': 25, 'Average_Score': 37.4}, {'Name': 'Jamez', 'Maths_score': 50, 'English_score': 50, 'Science_score': 53, 'Arts_score': 58, 'Economics_score': 60, 'Average_Score': 54.2}, {'Name': 'Joyce', 'Maths_score': 62, 'English_score': 68, 'Science_score': 58, 'Arts_score': 72, 'Economics_score': 60, 'Average_Score': 64.0}, {'Name': 'Janetti', 'Maths_score': 75, 'English_score': 73, 'Science_score': 78, 'Arts_score': 74, 'Economics_score': 80, 'Average_Score': 76.0}, {'Name': 'Joshua', 'Maths_score': 85, 'English_score': 45, 'Science_score': 56, 'Arts_score': 68, 'Economics_score': 51, 'Average_Score': 61.0}, {'Name': 'Joseph', 'Maths_score': 42, 'English_score': 25, 'Science_score': 35, 'Arts_score': 61, 'Economics_score': 20, 'Average_Score': 36.6}, {'Name': 'Jennifer', 'Maths_score': 63, 'English_score': 85, 'Science_score': 65, 'Arts_score': 82, 'Economics_score': 72, 'Average_Score': 73.4}, {'Name': 'Johnny', 'Maths_score': 62, 'English_score': 65, 'Science_score': 68, 'Arts_score': 64, 'Economics_score': 50, 'Average_Score': 61.8}]

ASSIGNING GRADES...

Joanna record shows performance as VERY GOOD

Justus record shows performance as EXCELLENT

Julius record shows performance as POOR - Work Harder

Jamez record shows performance as FAIR - Needs Improvement

Joyce record shows performance as GOOD

Janetti record shows performance as VERY GOOD

Joshua record shows performance as GOOD

Joseph record shows performance as POOR - Work Harder

Jennifer record shows performance as VERY GOOD

Johnny record shows performance as GOOD

	Name	Maths_score	English_score	Science_score	Arts_score	\
0	Joanna	75	68	92	72	
1	Justus	85	78	89	72	
2	Julius	35	55	40	32	
3	Jamez	50	50	53	58	
4	Joyce	62	68	58	72	
5	Janetti	75	73	78	74	
6	Joshua	85	45	56	68	
7	Joseph	42	25	35	61	
8	Jennifer	63	85	65	82	
9	Johnny	62	65	68	64	

	Economics_score	Average_Score	Grade	Remark
0	85	78.4	B	VERY GOOD
1	86	82.0	A	EXCELLENT
2	25	37.4	F	POOR - Work Harder
3	60	54.2	D	FAIR - Needs Improvement
4	60	64.0	C	GOOD
5	80	76.0	B	VERY GOOD
6	51	61.0	C	GOOD
7	20	36.6	F	POOR - Work Harder
8	72	73.4	B	VERY GOOD
9	50	61.8	C	GOOD

The Above Is The Exam Record Of The Selected Student And The Grades

Part 3: Add Descriptive Stats

```
In [141... # Import necessary libraries
import pandas as pd

# Using functions from the pandas module, find some descriptive information from the summary table
Overall_Class_Avg = stud_summ_table['average_score'.title()].mean()
Top_Score = stud_summ_table['average_score'.title()].max()
Top_student = stud_summ_table[stud_summ_table['average_score'.title()] == Top_Score]['name'.title()].tolist()
Least_score = stud_summ_table['average_score'.title()].min()
Worst_student = stud_summ_table[stud_summ_table['average_score'.title()] == Least_score]['name'.title()].tolist()

print('The Class Average is: {} \n'.format(Overall_Class_Avg))
print('The Top Student in the class is {}, with average score of {}\n'.format(Top_student[0], Top_Score))
print('The Lowest Student in the class is {}, with average score of {}\n'.format(Worst_student[0], Least_score))

# For the grade distribution
grade_order = ['A', 'B', 'C', 'D', 'F']
grade_distribution = stud_summ_table['grade'.title()].value_counts().reindex(grade_order, fill_value=0)
print('This is the Grade Distribution: {}'.format(grade_distribution))
```

The Class Average is: 62.48

The Top Student in the class is Justus, with average score of 82.0

The Lowest Student in the class is Joseph, with average score of 36.6

This is the Grade Distribution: Grade

A 1

B 3

C 3

D 1

F 2

Name: count, dtype: int64

In []: