



SRINIVASA EDUCATIONAL SOCIETY'S

PACE INSTITUTE OF TECHNOLOGY & SCIENCES **(AUTONOMOUS)**

Approved by AICTE, UGC, New Delhi & Govt. of Andhra Pradesh | Permanently Affiliated to JNTUK, Kakinada, A.P.

ACCREDITED BY NAAC WITH 'A' GRADE | ACCREDITED BY NBA

An ISO 9001 : 2008 Certified Institution | 'A' Grade Engineering College by Government of A.P.

NH-16, Near Valluramma Temple, ONGOLE - 523 272, A.P., Contact No.: 08592 278315, 9581456310 | www.pace.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING **(2023-2024)**

Internship and mini project based on python programming with Data Engineer

Project title: Scholastic Achievement Dashboard

Under the mentorship of
M.Shravan kumar

In Accordance with requirement of degree of
BACHELOR OF TECHNOLOGY

In
ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by:

CH.BHAVANAA
21KQ1A0203

Date: 11-06-2024

Scholastic Achievements Dashboard

Abstract:

This project is about the Attainment of students marks list. To store the data of student's name, roll number, section, subject name and marks for each student. Which provides the complete analysis of students marks with respect to the subjects.

Description:

This marks list contains the data of each student according to their roll number for each subject and Average marks. It also contains the pass percentage and fail percentage of the students with respect to the absentees in the class. It displays the subject name, subject code and date of examination along with the students data. Usually attainments are done in excel sheets but here we are going to make it using the python code.

Requirements:

1. **Selecting the Code:** Copy your Python code from your preferred editor or IDE.
2. **Opening Microsoft Word:** Open Microsoft Word on your computer.
3. **Pasting the Code:** Paste the copied Python code into the Word document.
4. **Formatting:** Format the code in the Word document to make it more readable. You can do this by selecting the pasted code and

applying the "Code" or "Preformatted Text" style from the Styles section in Word. This will maintain the formatting of the code.

5. **Adding Headers:** You might want to add headers to different sections of your code to make it more organized and understandable. For example, you could add headers like "Function Definitions", "Main Program", "Data Processing", etc.
6. **Saving the Document:** Once you've formatted the code and added any necessary headers or comments, save the Word document.
7. **Sharing or Printing:** You can now share the Word document with others or print it out as needed.

Approach:

• Initialization:

- Two empty lists `k` and `s` are initialized to store student information and total marks respectively.
- An empty dictionary `subject_total` is initialized to store the total marks for each subject.

Code Overview:

- Provide a high-level overview of the code structure.
- It consists of functions, loops, and data structures.

Function Definitions:

- List and explain each function used in the code.
- Describe their purpose, parameters, and what they return.

Main Program:

- Describe the main flow of the program.
- Explain how user input is gathered and how functions are called to process the data.

Data Structures:

- Explain the data structures used, such as lists and dictionaries.

- Describe how they are utilized to store student information and calculate totals.

Input Processing:

- Detail how user input is processed.
- Explain how student details are collected and stored in the appropriate data structure.

Calculations:

- Describe how total marks are calculated for each student.
- Explain any other calculations performed, such as finding highest/lowest marks.

Output:

- Explain how the program displays the results.
- Describe the formatting used to present student information and calculated values.

Conclusion:

- Summarize the functionality and significance of the program.
- Mention any potential improvements or future enhancements.

Code Listing:

- Provide the complete Python code with appropriate formatting.
- Include line numbers for reference.

Program:

```

projectB.py - C:\Users\HP\AppData\Local\Programs\Python\Python312\projectB.py (3.12.3)
File Edit Format Run Options Window Help
k=[]
s=[]
subject_total = {}
def stu_marks(a,b,c,d,e):
    student = {
        'Name': a,
        'Roll': b,
        'Sub': c,
        'Marks': d,
        'TMarks': e
    }
    k.append(student)

n=int(input())
A=input("enter school name: ")
B=int(input("No.of students: "))
C=input("Section: ")
for i in range(n):
    a=input("enter student name: ")
    b=int(input("enter roll number: "))
    c=tuple(map(str,input("enter sub names: ").split()))
    d=list(map(int,input("enter marks: ").split()))
    e=sum(d)
    stu_marks(a,b,c,d,e)
print("School name: ",A)
print("No.of students: ",B)
print("Section: ",C)
print('\tName\tRoll.no\t\t\t\t\tsubjects\t\t\t\t\tMarks\t\tTotal marks')

```

```
File Edit Format Run Options Window Help
print('\tName\tRoll.no\t\t\tsubjects\t\t\t\tMarks\t\tTotal marks')
for i in range(n):
    print(k[i])

for _ in range(len(c)):
    subject=c[i]
    marks=d[i]
    subject_total[subject]=subject_total.get(subject,0) + marks
|
max_marks=max(k,key=lambda x:x['TMarks'])
print('\nhighest marks')
print("name:",max_marks['Name'])
print("roll:",max_marks['Roll'])
print("sub:",max_marks['Sub'])
print("marks:",max_marks['Marks'])
print("Tmarks:",max_marks['TMarks'])
min_marks=min(k,key=lambda x:x['TMarks'])
print('\nlowest marks')
print("name:",min_marks['Name'])
print("roll:",min_marks['Roll'])
print("sub:",min_marks['Sub'])
print("marks:",min_marks['Marks'])
print("Tmarks:",min_marks['TMarks'])

print('\nhighest subject marks ')
for subject, total_marks in subject_total.items():
    print(subject+":",total_marks)
```

Output:

```
project op.B.py - C:\Users\HP\AppData\Local\Programs\Python\Python312\project op.B.py (3.12.3)
File Edit Format Run Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\HP\AppData\Local\Programs\Python\Python312\project op.B.py
3
enter school name: Sri Chaitanya Techno School
No.of students: 30
Section: XA
enter student name: Shiva
enter roll number: 201
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 86 85 87 76 75 78
enter student name: Krishna
enter roll number: 203
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 86 87 84 85 88 89
enter student name: Vishnu
enter roll number: 202
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 91 92 93 94 95 97
School name: Sri Chaitanya Techno School
No.of students: 30
Section: XA
Name Roll.no subjects Marks Total marks
{'Name': 'Shiva', 'Roll': 201, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [86, 85, 87, 76, 75, 78], 'TMarks': 487}
{'Name': 'Krishna', 'Roll': 203, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [86, 87, 84, 85, 88, 89], 'TMarks': 519}
{'Name': 'Vishnu', 'Roll': 202, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [91, 92, 93, 94, 95, 97], 'TMarks': 562}

highest marks
name: Vishnu
roll: 202
sub: ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc')
marks: [91, 92, 93, 94, 95, 97]
Tmarks: 562

lowest marks
name: Shiva
roll: 201
```

```
project op.B.py - C:\Users\HP\AppData\Local\Programs\Python\Python312\project op.B.py (3.12.3)
File Edit Format Run Options Window Help

enter roll number: 201
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 86 85 87 76 75 78
enter student name: Krishna
enter roll number: 203
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 86 87 84 85 88 89
enter student name: Vishnu
enter roll number: 202
enter sub names: Tel Hin Eng Mat Sci Soc
enter marks: 91 92 93 94 95 97
School name: Sri Chaitanya Techno School
No.of students: 30
Section: XA
Name Roll.no subjects Marks Total marks
{'Name': 'Shiva', 'Roll': 201, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [86, 85, 87, 76, 75, 78], 'TMarks': 487}
{'Name': 'Krishna', 'Roll': 203, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [86, 87, 84, 85, 88, 89], 'TMarks': 519}
{'Name': 'Vishnu', 'Roll': 202, 'Sub': ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc'), 'Marks': [91, 92, 93, 94, 95, 97], 'TMarks': 562}

highest marks
name: Vishnu
roll: 202
sub: ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc')
marks: [91, 92, 93, 94, 95, 97]
Tmarks: 562

lowest marks
name: Shiva
roll: 201
sub: ('Tel', 'Hin', 'Eng', 'Mat', 'Sci', 'Soc')
marks: [86, 85, 87, 76, 75, 78]
Tmarks: 487

highest subject marks
Eng : 558

sorted total marks: [487, 519, 562]
```

Explanation:

1. Data Structures Initialization:

- Three empty lists are initialized: k to store student information dictionaries, s to store total marks of students, and subject_total to store the total marks of each subject.

2. stu_marks Function:

- This function takes five parameters: student name (a), roll number (b), subjects (c), marks (d), and total marks (e).
- It creates a dictionary student containing the student's details and appends it to the list k.

3. User Input:

- The program prompts the user to input the number of students, school name (A), number of students (B), and section (C).
- Then, it iterates n times (where n is the number of students) to collect details for each student:
 - Student name (a), roll number (b), subjects (c), and marks (d) are inputted.
 - Total marks (e) are calculated by summing the marks.
 - The stu_marks function is called to store the student's details.

4. Output:

- School details (name, number of students, section) are printed.
- A table header for displaying student details is printed.
- Each student's details are printed using the data stored in the list k.

5. Calculations:

- Total marks for each subject are calculated and stored in the subject_total dictionary.
- The student with the highest total marks (max_marks) and the one with the lowest total marks (min_marks) are identified.
- The highest marks for each subject are printed.

6. Sorting:

- The total marks of all students are appended to the list s.
- The list s is sorted in ascending order.

7. **Output** (continued):

- The student with the highest total marks (max_marks) and the one with the lowest total marks (min_marks) are printed.
- The total marks of each subject are printed.
- The sorted total marks of all students are printed.

Conclusion:

In conclusion, the provided Python code offers a basic yet functional system for managing student marks within a school. It allows users to input student details including names, roll numbers, subjects, and marks, then calculates the total marks for each student and aggregates subject-wise totals. The program also identifies students with the highest and lowest total marks, as well as the highest marks in each subject. Finally, it sorts and displays the total marks of all students.

While this code provides a solid foundation for a student marks management system, there are several areas for potential improvement and expansion. These include:

1. **Error Handling:** Incorporating error handling mechanisms to handle unexpected user inputs or invalid data entries would enhance the robustness of the program.
2. **Modularity:** Breaking down the code into more modular functions could improve readability and maintainability, making it easier to extend and modify in the future.
3. **Data Persistence:** Implementing functionality to store student data persistently, such as in a database or file system, would enable data retrieval and manipulation across multiple sessions.

4. **User Interface:** Developing a graphical user interface (GUI) could enhance the user experience by providing a more intuitive and interactive way to input and view student data.
5. **Reporting:** Adding functionality to generate detailed reports, such as student performance summaries or subject-wise analysis, would provide valuable insights for educators and administrators.

Overall, while the current code fulfills its basic objectives, further enhancements and refinements could make it a more comprehensive and valuable tool for managing student marks in educational settings.

