

DAY 1 – PYTHON SPRINT

Core Topic: Lists, Tuples, Dictionary & sets

Resource: <https://www.youtube.com/watch?v=UrsmFxEIp5k> from 1hr 50 minutes till 2hrs 53 minutes

[Expected Time to Finish – 2-3 hours]

Core Problem: Personal Profile Processor

Everyone works on the same problem. Difficulty depends on how deep you take it. Choose your level and complete it properly.

BEGINNER LEVEL (Starting from Scratch)

Write a Python program that does the following:

1. Take name and age from the user.
2. Take 3 favorite subjects and store them in a list.
3. Take roll number and branch and store them in a tuple.
4. Take 3 subject-marks pairs and store them in a dictionary.
5. Take 3 technical skills and store them in a set.
6. Print all collected information.

Expected Time to Finish – 20 min

Sample Input:

```
Enter your name: Arshad Ali
Enter your age: 18
Enter subject 1: Physics
Enter subject 2: CS
Enter subject 3: Mathematics
Enter roll number: 02
Enter branch: Computer Engineering (Original)
Enter marks of Physics: 69
Enter marks of CS: 69
Enter marks of Mathematics: 69
Enter skill 1: Python
Enter skill 2: WebDev
Enter skill 3: Nahi aata
```

Sample Output:

```
--- Student Details ---
Name: Arshad Ali
Age: 18
Favorite Subjects: ['Physics', 'CS', 'Mathematics']
Student Info: ('02', 'Computer Engineering (Original)')
Marks: {'Physics': 69, 'CS': 69, 'Mathematics': 69}
Technical Skills: {'Nahi aata', 'WebDev', 'Python'}
```

INTERMEDIATE LEVEL

Student Info Analyzer (Comma-Separated Input Version)

Objective

To take user input and store data using **strings, lists, tuples, dictionaries, and sets**, and then **display and process** the stored information.

Task Description

Write a Python program that performs the following steps:

Input Section

1. Take the user's **full name** and format it in **title case**.
 2. Take the user's **roll number**.
 3. Take the user's **branch**.
 4. Take **three favorite subjects** as **comma-separated values**.
 5. Ask the user to enter **marks for each subject one by one**.
 6. Take **technical skills** as **comma-separated values**.
-

Data Storage Requirements

- Store the full name in a **string**.
- Store favorite subjects in a **list**.
- Store roll number and branch together in a **tuple**.
- Store subject-wise marks in a **dictionary**
(Subject → Marks).
- Store technical skills in a **set**
(duplicates should be removed automatically).

Processing Tasks

1. Display the **favorite subjects in alphabetical order.**
 2. Calculate and display:
 - o **Total marks**
 - o **Average marks**
 3. Display the **subject with the highest marks.**
-

Output Section

Display the following clearly:

- Full Name (formatted)
- Roll Number and Branch
- Favorite Subjects (sorted alphabetically)
- Subject-wise Marks
- Total Marks
- Average Marks
- Highest Scoring Subject
- Technical Skills

Expected Time to Finish – 30 min

Sample Input:

```
Enter your full name: Arshad Ali
Enter roll number: 02
Enter branch: Computer Engineering
Enter your favorite subjects (comma-separated):
Physics, Maths, CS
Enter marks of Physics: 69
Enter marks of Maths: 67
Enter marks of CS: 67
Enter technical skills (comma-separated): Python,
DS, Cpp, Web Dev
```

Sample Output:

```
--- Student Details ---
Full Name: Arshad Ali
Roll Number & Branch: ('02', 'Computer Engineering')
Favorite Subjects (Alphabetical): ['CS', 'Maths', 'Physics']
Subject-wise Marks: {'Physics': 69, 'Maths': 67, 'CS': 67}
Total Marks: 203
Average Marks: 67.666666666666667
Highest Scoring Subject: Physics
Technical Skills: {'Python', 'Web Dev', 'Cpp', 'DS'}
```

ADVANCED LEVEL MINI PROJECT

Build a CLI-Based Student Performance Tracker

Objective

You will create a Python program that allows users to **add, view, and analyze student data** via a **menu-driven command-line interface**.

Step 1: Plan the Data

Decide how you want to store each student's information. Each student should have:

- Full Name (string, formatted in title case)
- Roll Number (string or number)
- Branch (string)
- Favorite Subjects (3 subjects, comma-separated, store in a list)
- Marks for each subject (dictionary: subject → marks)
- Technical Skills (comma-separated, store in a list or set, remove duplicates if convenient)

Hint: You can store all students in a list of dictionaries.

Step 2: Build a CLI Menu

Your program should show a **menu with options**:

1. **Add Student** – Input all student details.
2. **View All Students** – Display all information for all students.
3. **Analytics** – Compute top students, subject-wise toppers, most common skills.
4. **Exit** – Quit the program.

Use a **while loop** to keep showing the menu until the user chooses Exit.

Step 3: Add Student Functionality

- Ask the user to enter **full name, roll number, branch**.
- Take **3 favorite subjects** as comma-separated input.
- Ask for **marks of each subject individually**.
- Ask for **technical skills** as comma-separated input.
- Store this information in a **student object** (dictionary or any suitable structure).
- Append the student to the **main students list**.

Step 4: View Students

For each student, display:

- Full Name
 - Roll Number & Branch
 - Favorite Subjects (sorted alphabetically if possible)
 - Marks for each subject
 - Total Marks
 - Average Marks
 - Highest Scoring Subject
 - Technical Skills
-

Step 5: Analytics

- Rank students by **total marks**.
 - Display **subject-wise toppers** (who scored highest in each subject).
 - Show **most common technical skills** among all students.
-

Step 6: Input Validation (Optional but Recommended)

- Marks should be **numbers** (preferably 0–100).
 - Ensure **roll numbers are unique**.
 - Optional: Remove duplicate skills automatically.
-

Step 7: Loop Until Exit

- After performing any operation (Add/View/Analytics), **return to the menu**.
- Exit only when the user chooses the **Exit option**.

Step 8: Optional Enhancements

- Search for students by **name, roll number, or branch**.
 - Filter students by **skill or total marks range**.
 - Export data to a **file** (JSON or text).
-

Tips for Students

- Use functions to organize your code (e.g., `add_student()`, `view_students()`, `analytics()`).
- Use **lists, dictionaries, sets** as needed.
- Use `sorted()` for sorting subjects or skills.

- Use max() with key for finding top students or highest scoring subject.
- Keep the **CLI interface user-friendly**.

Expected Time to Finish – 3-4 hours

Sample Run:

```

--- Student Performance Tracker ---
1. Add Student
2. View All Students
3. Analytics
4. Exit
Enter your choice: 1

--- Add New Student ---
Enter full name: Arshad Ali
Enter roll number: 101
Enter branch: CSE
Enter 3 favorite subjects (comma-separated): Math, Physics, CS
Enter marks of Math: 95
Enter marks of Physics: 88
Enter marks of CS: 92
Enter technical skills (comma-separated): Python, Git, Python

Arshad Ali added successfully!

--- Student Performance Tracker ---
1. Add Student
2. View All Students
3. Analytics
4. Exit
Enter your choice: 1

--- Add New Student ---
Enter full name: Broken User
Enter roll number: 102
Enter branch: IT
Enter 3 favorite subjects (comma-separated): English, CS, Biology
Enter marks of English: 78
Enter marks of CS: 85
Enter marks of Biology: 80
Enter technical skills (comma-separated): Java, Python, Java

Broken User added successfully!

--- Student Performance Tracker ---
1. Add Student
2. View All Students
3. Analytics
4. Exit
Enter your choice: 2

--- All Students ---

Name: Arshad Ali
Roll & Branch: 101, CSE
Favorite Subjects (sorted): ['CS', 'Math', 'Physics']
Marks: {'Math': 95.0, 'Physics': 88.0, 'CS': 92.0}
Total Marks: 275.0, Average: 91.67
Highest Scoring Subject: Math
Technical Skills: ['Python', 'Git']

Name: Broken User
Roll & Branch: 102, IT
Favorite Subjects (sorted): ['Biology', 'CS', 'English']
Marks: {'English': 78.0, 'CS': 85.0, 'Biology': 80.0}
Total Marks: 243.0, Average: 81.00
Highest Scoring Subject: CS
Technical Skills: ['Java', 'Python']

--- Student Performance Tracker ---
1. Add Student
2. View All Students
3. Analytics
4. Exit
Enter your choice: 3

--- Top Students by Total Marks ---
1. Arshad Ali - Total Marks: 275.0
2. Broken User - Total Marks: 243.0

--- Subject-wise Toppers ---
Math: Arshad Ali (95.0 marks)
Physics: Arshad Ali (88.0 marks)
CS: Arshad Ali (92.0 marks)
English: Broken User (78.0 marks)
Biology: Broken User (80.0 marks)

--- Most Common Technical Skills ---
Python: 2 student(s)
Git: 1 student(s)
Java: 1 student(s)

--- Student Performance Tracker ---
1. Add Student
2. View All Students
3. Analytics
4. Exit
Enter your choice: 4

Exiting program...

```

Submission & Deadlines

Deadline: Tomorrow by 9:15 PM.

Format: Submit via WhatsApp using a link to GitHub !!, or by pasting the code (not recommended)