□ Low-Level Design (LLD) – Student Performance Tracker

Difficulty Level: Easy | **Total Marks:** 20
Standards Followed: 4 Functions | 4 Visible Test Cases

□ Summary of Corrections (Based on SME Feedback)

- □ Used Pandas functions for groupby and filtering
- □ Each function structured to run independently
- □ Output types match test expectation
- □ Included validation for scores

□ Concepts Tested
□ CSV Loading using Pandas
□ Conditional Filtering
□ Aggregation and Grouping
□ Mean and Sorting Functions

□ Problem Statement
You are given a CSV file that records student marks across different subjects.
Your task is to analyze academic performance using Pandas.

□ Operations

□ 1. Load Student Data
□ Loads the CSV file into a DataFrame.

□ Function Prototype:

```
def load_student_data(file_path: str) -> pd.DataFrame:
```

□ Input: `"marks.csv"`
□ Output: DataFrame

□ Implementation Flow:

- Use `pd.read_csv()` to read the CSV
- Return the full DataFrame

□ **2. Get Top Scorers**

□ Return students with scores above 90 in any subject.

□ Function Prototype:

```
def get_top_scorers(df: pd.DataFrame) -> pd.DataFrame:
```

□ Input: DataFrame
□ Output: DataFrame filtered with at least one score > 90

□ Implementation Flow:

- Check for each row if any subject score > 90
- Return filtered DataFrame

□ **3. Average Score per Subject**

□ Calculate average marks for each subject.

□ Function Prototype:

```
def average_scores(df: pd.DataFrame) -> dict:
```

□ Input: DataFrame
□ Output: Dictionary – {subject: average_score, ...}

□ Implementation Flow:

- Use `df.mean()` on subject columns
- Convert to dictionary and return

□ **4. Top N Students by Total Score**

□ Return top 3 students based on total marks.

□ Function Prototype:

```
def top_students(df: pd.DataFrame) -> list:
```

□ Input: DataFrame
□ Output: List of tuples – [(student_name, total_score), ...]

□ Implementation Flow:

- Sum subject scores row-wise
- Sort and get top 3
- Return name and score in tuple

□ Implementation Hints

```
# □ Implementation Hints for Student Performance Tracker
import pandas as pd

class StudentPerformanceTracker:

    def load_student_data(self, file_path: str) -> pd.DataFrame:
        pass  # TODO

    def get_top_scorers(self, df: pd.DataFrame) -> pd.DataFrame:
        pass  # TODO

    def average_scores(self, df: pd.DataFrame) -> dict:
        pass  # TODO

    def top_students(self, df: pd.DataFrame) -> list:
        pass  # TODO
```

□ Test Cases & Marks Allocation

Test Case ID	Description	Associated Function	Marks
TC1	Load CSV correctly	load_student_data()	□ 5
TC2	Get high-performing students	get_top_scorers()	□ 5
TC3	Calculate subject-wise averages	average_scores()	□ 5
TC4	List top 3 students by total score	top_students()	□ 5
	Total Marks	–	□ 20

□ Visible Test Cases

□ TC1: Load Student CSV

```
python

Input: "marks.csv"
Expected Output: DataFrame with 5 columns and 10+ rows
```

□ TC2: Top Scorers Filter

```
get_top_scorers(df)
Expected Output: Students with any subject > 90
```

□ TC3: Subject Averages

```
average_scores(df)
Expected Output: {'Math': 78.5, 'Science': 82.2, 'English': 80.1}
```

□ TC4: Top 3 by Total Marks

```
top_students(df)
Expected Output: [("Alice", 287), ("John", 275), ("Sara", 270)]
```