Industry Problem Statement

Student Academic & Scholarship Management System (Python)

Business Background

A university wants to build a Python-based internal system to manage:

- Student academic records
- Subject-wise marks
- GPA calculation
- Scholarship eligibility and awards

You are required to implement this system **incrementally**, following **real academic ERP logic** and clean coding practices.

Task 1: Capture Student Details (Input Validation)

Objective

Collect and store basic student information.

Requirements

Write a program to accept:

- Student ID
- Student Name
- Course Name

Business Rules

- Student ID must not be empty
- Name must contain only alphabets

Expected Outcome

Validated student record ready for academic processing.

Task 2: Capture Subject Marks

Objective

Record academic performance data.

Requirements

- Accept marks for **5 subjects**
- Store marks in a list

Business Rules

- Marks must be between **0 and 100**
- Reject invalid entries

Expected Outcome

Clean list of subject marks.

Task 3: Total Marks Calculation

Objective

Compute aggregate academic score.

Requirements

- Calculate total marks from the marks list

Expected Outcome

Total marks used for percentage and GPA calculation.

Task 4: Percentage Calculation

Objective

Convert total marks into percentage.

Formula

```
Percentage = (Total Marks / 500) × 100
```

Expected Outcome

Accurate percentage score.

Task 5: GPA Calculation

Objective

Standardize academic performance.

Rules

- GPA = Percentage ÷ 10
- Round GPA to **2 decimal places**

Expected Outcome

Valid GPA score.

Task 6: Academic Classification

Objective

Categorize academic standing.

Rules

GPA	Classification
≥ 9.0	Distinction
≥ 7.5	First Class
≥ 6.0	Second Class
< 6.0	Fail

Expected Outcome

Readable academic classification.

Task 7: Scholarship Eligibility Check

Objective

Identify students eligible for scholarships.

Rules

- GPA ≥ 8.5

- No subject marks below 60

Expected Outcome

Clear eligibility decision (Yes / No).

Task 8: Scholarship Amount Calculation

Objective

Determine financial assistance.

Rules

- Eligible students receive ₹50,000 scholarship
- Others receive ₹0

Expected Outcome

Scholarship amount.

Task 9: Store Student Academic Records

Objective

Simulate database storage.

Requirements

- Store student data in a list of dictionaries

Expected Outcome

Structured academic data collection.

Task 10: Academic Summary Report (Procedural)

Objective

Generate a student academic summary.

Summary Should Include

- Student ID & Name

- GPA
- Classification
- Scholarship Status

Task 11: Student Class Design (OOP)

Objective

Model student as an object.

Create class `Student` with:

Attributes

- student_id
- name
- course
- marks

Task 12: Academic Methods

Objective

Encapsulate academic logic.

Methods

- calculate_percentage()
- calculate_gpa()

Task 13: Classification Method

Objective

Encapsulate grading logic.

Method

- get_classification()

Task 14: Scholarship Method

Objective

Automate scholarship processing.

Method

- calculate_scholarship()

Task 15: Final Report Card Generation

Objective

Generate a professional student report card.

Output Format (Example)

```
Student ID     : S101
Name           : Rahul Sharma
Course         : B.Tech
GPA            : 8.7
Classification : First Class
Scholarship    : ₹50,000
```