Project Report: Relative Grading System using Python

1. Introduction

This project implements a relative grading calculator using basic Python concepts. Universities often use relative grading to evaluate student performance based on the distribution of marks. Instead of absolute cutoffs, this system uses the mean and standard deviation of the class to determine grade boundaries. The program developed here automates this grading method.

2. Objective

The main objective is to create a Python program that can:

- Accept inputs for total marks, mean, SD, and internal/TEE marks
- Use clearly defined relative grade thresholds
- Output both the grade and the corresponding grade point
- Demonstrate the use of basic Python programming constructs

3. Concepts Used

This project is intentionally built using only fundamental Python concepts:

- Variables and arithmetic expressions
- User input (input())
- Conditional statements (if-elif-else)
- Functions
- Dictionaries for grade-point mapping
- Printing output to the console

These concepts make it ideal for beginners and educational use.

4. Problem Definition

Relative grading evaluates a student based on their performance compared to the class average. Manual calculation of relative thresholds (like mean + 1.5 SD) is time-consuming and error-prone. This Python program automates these calculations and ensures consistent results.

5. Methodology

The program follows these steps:

1. Accepts input: total marks, mean, standard deviation, CAT1 marks, CAT2 marks, and TEE marks.
2. Compares the total marks against relative thresholds:

- S for ≥ mean + 1.5 SD
- A for ≥ mean + 0.5 SD
- B for ≥ mean – 0.5 SD
- C for ≥ mean – 1.0 SD
- D for ≥ mean – 1.5 SD

3. If none of the above apply, the program checks pass conditions:
   - TEE ≥ 40%
   - Overall ≥ 40%
4. If pass conditions fail → grade = F
5. Grade is mapped to grade point using a dictionary
6. The program outputs both values

This method follows widely used university grading standards.

6. Implementation

A function relative_grade() computes the grade and grade point. The main program handles inputs and displays results. The code is simple, modular, and easy to modify.

7. Results

The program outputs:

- The relative grade (S/A/B/C/D/E/F)
- The numerical grade point

This helps students quickly estimate their academic performance relative to class statistics.

8. Applications

- Academic evaluation
- Classroom grading analysis
- Student performance prediction
- Useful for universities using relative grading systems

9. Conclusion

The Python-based relative grading system provides a simple and efficient way to compute grades using mean and standard deviation. It is accurate, easy to use, and demonstrates practical application of basic Python

programming concepts. The project can be expanded further by adding file storage, GUIs, or automated data import in future versions.