



# Assignment 2 – GradeBook Analyzer

**Course:** Programming for  
Problem Solving using  
Python

**Student:** Arpit Patni

**Roll No:** 2501730111

**Program:** B.Tech CSE  
(AIML), Section A

**Submitted To:** Dr. Sameer  
Farooq

# 1. Introduction

The GradeBook Analyzer is a Python-based system developed to simplify the evaluation of student performance. In academic settings, teachers often need to calculate averages, assign grades, track pass and fail outcomes, and prepare structured summaries. Doing these tasks manually is time-consuming and error-prone.

This assignment demonstrates how Python can automate these processes through user-friendly input handling, data structures, conditional logic, and optional CSV export functionality. The project highlights the use of Python to build practical educational tools that improve accuracy and efficiency.

## 2. Aim of the Assignment

- To design a program that allows entry of student scores manually or through a CSV file.
- To compute essential statistics such as average, highest, and lowest marks.
- To generate grade categories (A, B, C, D, F) based on score thresholds.
- To segregate pass and fail lists.
- To optionally export the entire result dataset into a CSV file.
- To demonstrate the use of Python's data structures, conditional statements, loops, and file I/O.

## 3. Problem Statement

Develop a Python program that:

1. Accepts marks for multiple students.
2. Computes summary statistics (mean, maximum, minimum).
3. Assigns grades based on predefined grading criteria.
4. Displays grade distribution and pass/fail lists.
5. Provides the option to save results into a CSV file.

The system should be simple, interactive, and capable of handling any number of students.

## 4. Algorithm / Procedure

1. Start the program and ask the user how many students' marks will be entered.
2. For each student:
  - a. Ask for the student's name.
  - b. Ask for the marks obtained.
  - c. Store both values in lists or dictionaries.
3. After data input is complete:
  - a. Use max(), min(), and average formulas to compute statistics.
  - b. Assign grades according to score ranges.
  - c. Separate students into pass and fail groups.
4. Display the summary on screen:
  - a. Average marks
  - b. Highest marks
  - c. Lowest marks
  - d. Grade distribution
  - e. Pass/Fail lists
5. Ask the user whether to export data to CSV.
6. If yes, save all details into grade\_output.csv.
7. End the program.

## 5. Code Explanation

### Input Handling

The program uses simple input functions (`input()`) to collect names and marks. Marks are validated using numeric conversion (`int()` or `float()`).

### Data Storage

Two parallel lists or a list of dictionaries is used to store student information efficiently:

```
students = []
students.append({"name": name, "marks": marks})
```

### Statistics Calculation

- **Average** =  $\text{sum}(\text{marks}) / \text{total students}$

- **Maximum** = `max(marks)`
- **Minimum** = `min(marks)`

## Grade Assignment

Conditional statements decide grade categories:

```
if marks >= 90: grade = 'A'
```

```
elif marks >= 80: grade = 'B'
```

## Pass/Fail Check

Students scoring 40 or above are added to the *pass* list; otherwise to *fail*.

## CSV Export

Python's csv module is used to write all computed data into a CSV file:

```
with open("grade_output.csv","w") as f:
```

```
    writer = csv.writer(f)
```

```
    writer.writerow([...])
```

## Output Display

All summaries are printed in a structured format for clarity.

```

===== GRADEBOOK ANALYZER =====
1. Enter student data manually
2. Load student data from CSV
3. Exit
=====
Choose an option (1-3): 1
How many students? 5
Enter student name: Arpit
Enter marks for Arpit: 100
Enter student name: Vikas
Enter marks for Vikas: 90
Enter student name: Riya
Enter marks for Riya: 70
Enter student name: Mehul
Enter marks for Mehul: 90
Enter student name: Chakshu
Enter marks for Chakshu: 90

----- STATISTICAL SUMMARY -----
Average Score: 88.00
Median Score: 90.00
Highest Score: Arpit (100.0)
Lowest Score: Riya (70.0)

----- GRADE DISTRIBUTION -----
A: 4
B: 0
C: 1
D: 0
E: 0
F: 0

----- PASS / FAIL -----
Passed (5): ['Arpit', 'Vikas', 'Riya', 'Mehul', 'Chakshu']
Failed (0): []

      Name      Marks   Grade
-----
Arpit      100.0    A
Vikas      90.0     A
Riya       70.0     C
Mehul      90.0     A
Chakshu    90.0     A

```

Do you want to export results to CSV? (y/n): y  
Results exported to grade\_output.csv

## 6. Output Explanation

- Program displays a menu asking user for number of students.
- User enters student names and marks one-by-one.
- The program calculates average marks, the highest achiever, and lowest score.
- Grades are generated for each student.
- It shows:
  - Total students

- Grade-wise count
- Pass and fail lists
- When prompted, the user chooses to export results.
- Program generates: **grade\_output.csv** successfully and displays confirmation.

## 7. Conclusion

The GradeBook Analyzer successfully fulfills the objectives of Assignment-2 by providing an automated method to record student marks, compute essential statistics, generate grades, and evaluate overall performance. The CSV export option enhances usability by allowing teachers to save results for future reference.

This assignment strengthened understanding of Python lists, dictionaries, conditional logic, loops, and file handling. The project demonstrates Python's capability to create practical and efficient academic tools.

## 8. GitHub Links

### 🔗 Assignment-2 Repository:

<https://github.com/arpitpaatni2007-jpg/Python-assissgnments/tree/main/Assissgnment-2>

### 🔗 GitHub Profile:

<https://github.com/arpitpaatni2007-jpg>