Python Programming Lab
Exercise 1: Student Grading System

Department of Mechatronics Engineering

October 13, 2025

---

1 Objectives

The objective of this exercise is to integrate and apply all the concepts learned in:

- Lab 01: Variables, Data Types, Input/Output, Arithmetic Operators, Comments.
- Lab 02: Conditional Statements, Logical Operators, Lists, Loops.

Students will design and implement a complete Python program that:

- Accepts multiple student records.
- Calculates total and average marks.
- Determines the grade based on predefined criteria.
- Computes class statistics.

2 Problem Description

You are required to write a Python program to build a simple **Student Grading System** for a small class.

Problem Requirements

Step 1: Ask the user to input the **number of students**.

Step 2: For each student:

- Read the **name** of the student.
- Read **five subject marks** (out of 100).

Step 3: Store the marks in a **list**.

Step 4: Calculate:

- Total marks.
- Average mark.
- Grade according to:
  - A: average ≥ 90
  - B: $80 \leq$ average < 90
  - C: $70 \leq$ average < 80
  - D: $60 \leq$ average < 70
  - F: average < 60

Step 5: Display:

- Student name.
- Entered marks.
- Total, average, and grade.

Step 6: After all students:

- Compute class average.
- Find highest and lowest average.

Step 7: Ask the user if they want to enter another class.

3 Concepts Used

- **Variables and Data Types**: to store names, marks, totals, averages.
- **Arithmetic Operations**: total and average calculations.
- **Input and Output**: for interacting with the user.
- **Conditional Statements**: to assign letter grades.
- **Logical Operators**: used in grade classification.
- **Lists**: for storing multiple subject marks.
- **Loops**: to process each student and each subject.

- **Built-in Functions**: `sum()`, `max()`, `min()`, `len()`.

4 Sample Execution

```
Enter number of students: 2

--- Student 1 ---
Enter student name: Ahmed
Enter mark for subject 1: 90
Enter mark for subject 2: 85
Enter mark for subject 3: 92
Enter mark for subject 4: 88
Enter mark for subject 5: 91
Total = 446, Average = 89.20, Grade = B

--- Student 2 ---
Enter student name: Sara
Enter mark for subject 1: 78
Enter mark for subject 2: 81
Enter mark for subject 3: 85
Enter mark for subject 4: 74
Enter mark for subject 5: 80
Total = 398, Average = 79.60, Grade = C

Class average: 84.40
Highest average: 89.20
Lowest average: 79.60

Do you want to enter another class? (yes/no): no
Goodbye!
```

5 Solution Code

```
# ============================================
# Student Grading System - Final Exercise
# ============================================

while True:
num_students = int(input("Enter number of students: "))

# Lists to store student data
student_names = []
student_averages = []

for i in range(num_students):
print(f"\n--- Student {i+1} ---")
name = input("Enter student name: ")
marks = []
```

```
        # Input 5 marks for each student
        for j in range(5):
        mark = float(input(f"Enter mark for subject {j+1}: "))
        marks.append(mark)

        total = sum(marks)
        average = total / len(marks)

        # Determine grade
        if average >= 90:
        grade = "A"
        elif average >= 80:
        grade = "B"
        elif average >= 70:
        grade = "C"
        elif average >= 60:
        grade = "D"
        else:
        grade = "F"

        print(f"Total = {total}, Average = {average:.2f}, Grade =
            {grade}")

        student_names.append(name)
        student_averages.append(average)

        # Class statistics
        class_average = sum(student_averages) / len(
           student_averages)
        highest = max(student_averages)
        lowest = min(student_averages)

        print("\n================================")
        print(f"Class average: {class_average:.2f}")
        print(f"Highest average: {highest:.2f}")
        print(f"Lowest average: {lowest:.2f}")
        print("================================")

        again = input("Do you want to enter another class? (yes/
           no): ").lower()
        if again != "yes":
        print("Goodbye!")
        break
```

6 Remarks and Extensions

- You can extend this program to:
  - Save results to a file.

  - Use functions to organize the code.
  - Add input validation.
  - Display class ranking.

- This exercise gives students the first experience of integrating multiple Python programming elements into a structured program.