Minor Project Guidelines – Python & NumPy (Week-Off Assignment)

1. Objective of the Minor Project

The purpose of this minor project is to help students: - Apply **basic Python and NumPy concepts** learned so far - Build **logical thinking and problem-solving skills** - Gain **confidence in writing code independently** - Understand how small programs are structured end-to-end

This project is intentionally kept **simple and limited in scope**, but it must be implemented **entirely using the student's own logic**.

---

2. Important Rules and Academic Integrity

Students must strictly follow these rules:

1. **Do not use ChatGPT or any AI tools** to generate code.
2. You may refer to:
3. Your **class notes**
4. **Python official documentation**
5. Previously written practice programs
6. Direct copy-paste from the internet is **not allowed**.
7. Every student must be able to **explain their code line by line**.
8. Code similarity checks may be performed.

The goal is **learning**, not perfection.

---

3. Allowed Topics Only

Students are allowed to use **only** the following topics:

Python Basics

- Variables and data types
- `input()` and `print()`
- Arithmetic operators
- Conditional statements (`if`, `elif`, `else`)
- `while` and `for` loops
- Functions
- Lists

NumPy

- `numpy.array`
- Basic array operations
- Mathematical operations on arrays
- `sum()`, `mean()`, `max()`, `min()`
- Simple indexing and slicing

Using advanced topics beyond this list is **not permitted**.

---

4. Project Option 1: Console-Based Smart Calculator

4.1 Problem Statement

Create a **menu-driven calculator** that performs multiple mathematical operations using Python and NumPy.

4.2 Functional Requirements

The calculator should: 1. Display a menu with operations 2. Take user choice as input 3. Ask for required numbers 4. Perform calculation 5. Display result 6. Allow the user to continue or exit

4.3 Mandatory Operations

- Addition
- Subtraction
- Multiplication
- Division (handle division by zero)
- Power calculation
- Modulus operation

At least **one operation must use NumPy arrays**.

4.4 Sample Flow (Logic Only)

```
1. Show menu
2. Take choice
3. If choice is arithmetic:
   - Ask how many numbers
   - Store numbers in a list or NumPy array
   - Perform operation
4. Print result
5. Ask user to continue
```

Students must write their **own logic**, not follow this flow blindly.

---

5. Project Option 2: Student Marks Analyzer (Recommended)

5.1 Problem Statement

Develop a program that analyzes student marks using Python and NumPy.

5.2 Functional Requirements

The program should: - Accept marks for multiple students - Store marks in a NumPy array - Calculate: - Total marks - Average marks - Highest and lowest marks - Assign grade based on average

5.3 Grade Logic (Student-Defined)

Students must define their own grading rules such as: - A: Above 85 - B: 70–85 - C: 50–69 - Fail: Below 50

Grades must be implemented using **conditional statements**.

---

6. Project Option 3: Daily Expense Tracker

6.1 Problem Statement

Create a program to track daily expenses and analyze spending.

6.2 Functional Requirements

- Take daily expenses as input
- Store values in a NumPy array
- Calculate:
- Total expense
- Average expense
- Highest expense day
- Display spending summary

Optional: - Warn user if average expense exceeds a limit

---

7. Mandatory Implementation Guidelines

Every project **must include**:

1. At least **one user-defined function**
2. Use of **lists and NumPy arrays**
3. At least **one loop**
4. At least **one conditional block**

5. Meaningful variable names
6. Proper indentation and readability

---

8. Submission Requirements

Students must submit:

1. One **Google Colab notebook (.ipynb)**
2. Notebook must contain:
3. Project title
4. Problem statement (in student's own words)
5. Logic explanation (brief)
6. Complete code
7. Sample output

---

9. Evaluation Criteria

Criteria	Weightage
Logic and correctness	40%
Proper use of Python & NumPy	30%
Code readability	15%
Explanation and clarity	15%

---

10. Final Note to Students

This project is designed to: - Make you **think independently** - Strengthen your **programming fundamentals** - Prepare you for larger projects in the future

Focus on **logic first**, then code.

End of Guidelines.