

## Final Project Guidelines: Programming Logic and Design

### Project Title:

Python-Based Simple Application Development

### Project Overview:

For the final project in Programming Logic and Design, students will work collaboratively to develop a simple application using Python that demonstrates their understanding of programming fundamentals, structured logic, and problem-solving. The project aims to integrate key concepts of the course — including algorithm design, pseudocode creation, and flowchart representation — into a functional program that performs a specific real-world task.

### Group Composition:

- Each group must consist of 4–5 members.
- Every member should contribute actively to the design, coding, and documentation phases.
- Suggested roles include:
  - Team Leader – oversees project progress and task distribution
  - Programmer(s) – writes and tests Python code
  - System Designer – creates flowcharts and diagrams
  - Documenter – prepares written documentation
  - Presenter – leads the final presentation and demonstration

### Project Requirements:

#### 1. Application Concept

- Choose a simple, functional application that solves a real-world problem or performs useful tasks.  
Examples: Calculator, grading system, attendance tracker, password generator, quiz app, basic inventory system, etc.
- The project must include user input, processing, and output components.

#### 2. Implementation

- The application should be developed in Python using proper logic structures:
  - Sequential, conditional (if-else), and iterative (loops) statements
  - Functions and modular code organization
  - Basic use of lists, dictionaries, or other data types
- Include input validation and error handling.
- Use a console-based interface (GUI optional).

#### 3. Documentation

The project report should be clear, complete, and organized in the following format:

- A. Cover Page (Project title, group members, section, course, instructor, and date)
- B. Introduction / Project Overview
- C. Objectives
- D. Narrative Algorithm – step-by-step description of program logic
- E. Pseudocode – structured pseudocode of your Python program
- F. Flowchart – visual representation of program logic
- G. Program Code and Explanation
- H. Sample Input and Output
- I. Conclusion / Learning Reflection

#### 4. Presentation

- Each group will present and demonstrate their project to the class.
- Duration: 8–10 minutes per group.
- Presentation must include:
  - Brief explanation of the project concept
  - Demonstration of program execution
  - Discussion of algorithm, pseudocode, and flowchart

#### Evaluation Criteria:

Criteria	Description	Weight (%)
Functionality	The program runs correctly and meets objectives	25%
Logic Design	Correct use of algorithm, pseudocode, and flowchart	25%
Code Quality	Clarity, structure, and use of proper programming logic	20%
Documentation	Completeness and organization of the written report	15%
Presentation & Teamwork	Clear presentation and equal group participation	15%
Total		100%

#### Submission Guidelines:

- Submit a .zip folder named: PLD\_FinalProject\_Group#\_AppName
- Folder Contents:
  - Python source code (.py file)

- Documentation (Word or PDF format)
- Presentation slides (PPT or PDF)
- Deadline: [Insert Date Here]
- Late submissions will receive a 10% deduction per day.

### **Bonus Points (+5):**

Earn additional points by integrating:

- File handling (save/load data)
- Simple GUI (Tkinter or PyQt)
- Application of basic data structures