National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Faculty of Computing

CS 110: Fundamentals of Computer Programming
Assignment # 1

Assignment Date	13/10/2025	Due Date:	Sunday, 26/10/2025 (11:59 pm)	Marks: 30
CLO Attainment	[CLO-2]	Solve given real-world problem by applying appropriate programming concepts and techniques		
	[CLO-4]	Perform effectively as an individual and as a member of a team.		

Assignment Title: Delivery Drone Simulator

Learning Objectives:

By completing this assignment, you will:

- Strengthen **problem-solving and logic design** skills.
- Learn to break a program into clear **functions**.
- Experience **team coordination** using Git.
- Simulate **real-world decision making** with changing conditions.

Overview:

You are developing software to control a **delivery drone** that must deliver three packages in a day to:

- Location A
- Location B
- Location C

Before each flight, the drone must check:

- **Weather conditions** (sunny / windy / rainy)
- **Battery level**
- **Obstacle presence** (e.g., birds, restricted zone)

Based on these conditions, the drone decides whether to:

- Take off and deliver,
- Delay the delivery, or
- Return to base for recharge.

At day's end, display a **mission summary** (total deliveries, successful, failed, delayed, remaining battery, etc.).

Functional Requirements:

Your program must:

- **Start the Simulation**
  - Display a welcome message and initial battery (e.g., 100%).
  - Ask the user to begin the delivery day.
- **Environment Generation**
  - Randomly generate:
    - Weather (1 = sunny, 2 = windy, 3 = rainy)
    - Obstacle (0 = none, 1 = yes)
    - Battery drain per trip (between 10–25%)
  - Use rand() from <cstdlib> and srand(time(0)) for variability.
- **Decision Logic**
  - If rainy → no flight, mark as **delayed**.
  - If windy and battery < 40% → return to base to recharge (+10%).
  - If obstacle detected → reroute and add extra battery drain (+5%).
  - Otherwise, perform delivery successfully.
- **Modular Design**
  Suggested functions (but not limited to):
  - void startDay()
  - int getWeather()
  - bool checkObstacle()
  - bool deliverPackage(string location, int &battery)
  - void displaySummary(int success, int failed, int delayed, int battery)
- **Heavy Load Warning** (affects range or speed).
- **Optional Enhancements (for extra marks / curiosity)**
  - Random "system malfunction" event (10% chance) → delivery failed.
  - User choice to recharge mid-mission.
  - Introduce a performance score (based on success – battery used).
  - Add delay timers using loops for realism.

Team Roles:

This activity emphasizes **collaboration** and **problem-solving** while using tools like **Git/GitHub** for version control. You are free to choose/define roles for your assignment but they should be neatly identified (as shown in sample tables). Each group can have 2-3 group members including a group lead. **Make sure to select a team lead.**

Sample Roles	Responsibility
Logic Designer	Designs flowchart and logic conditions (weather, obstacle, battery).
Programmer 1	Implements environment generation and delivery decision functions.
Programmer 2	Implements user interaction, summary display, and testing.
Tester / Documenter	Tests all random conditions, maintains README.md, and summarizes AI tool use.

AI-based tools (e.g., ChatGPT, Copilot, Gemini) may be used **only for assistance**, not for direct code generation, and their use must be **acknowledged in the documentation**.

Concise Deliverables:

1. Source code (`.cpp`) (Code should be well-commented)

2. `README.md` with:

Section	What to Include
Title and Team Information	- Project title (Delivery Drone Simulator) - Team member names and roles (e.g., Logic Designer, Tester, Git Manager)
Overview / Problem Description	- Short summary of what the program does - Why this problem is interesting or real-world relevant
Program Design / Logic	- List of functions implemented - Brief explanation of how logic flows (e.g., "moveDrone() is called after each environment update") - Description of how random environmental factors are simulated
Execution Instructions	- How to compile and run the code (with sample input/output) - Any assumptions made
Team Collaboration Summary	- Description of how roles were divided - Evidence of Git usage (branch names, commit screenshots, etc.)
AI Tool Reflection (if any)	- Tools used (ChatGPT, Copilot, etc.) - What tasks they helped with (e.g., debugging, documentation structure) - A brief reflection on what students learned from using them
Future Improvements	- Two short points on what could be added if they had learned arrays or file handling or any other construct

3. GitHub Repo Link (each member must have at least one commit)

Evaluation Rubric (Total = 30 Marks)

Criterion	Marks	Low (1–2)	Average (3–4)	High (5)
1. Problem-Solving & Logic Design	5	Simple or inconsistent flow; major logic gaps	Reasonable modularity, few logical errors	Clear, dynamic flow; smooth interaction of random conditions and decisions
2. Team Coordination & Use of Git	5	Unequal contributions; minimal Git activity	Some collaboration, partial branching	Balanced collaboration; meaningful commits & merges
3. Documentation & Reflection	5	Missing/unclear README; no AI reflection	Basic README; limited AI notes	Detailed README; insightful reflection & clear project structure
4. Demo & Viva	5			

Submission Criteria:

This is a group assignment. Divide the tasks equally among each group member (all tasks have equal marks). Submit the assignment in the form of a report (pdf) on the LMS. The report should contain the following sections:

- Title page (provided on LMS)
- Table of contents
- Brief overview of the Requirements

- Contents of the Readme.md
- Test cases with screenshots of output
- GitHub Repo Link - Annex-I
- Code – Annex-II

All group members should submit the same document in LMS (Variations are not accepted).

Name the file using the given nomenclature:

FirstMemberName-SecondMemberName-ThirdMemberName-Assignment_1.docx.

Late submissions are accepted only till **Tuesday 28th October 2025** but with a penalty.

Plagiarism will be marked zero.