

CA2 — Pair Project (30%)

Module: DATA I8Z01 — Algorithms & Data Structures **Due:** See Moodle for date/time & presentation schedule.

Overview

In pairs, create a **teaching screencast** that explains **one** topic from either *Data Structures* or *Algorithm Paradigms* to an audience who can program but is new to the topic. Your screencast must: (1) explain *what/why/when*, (2) **demonstrate** key ideas with C++ code and worked examples, and (3) **analyze performance** using Big-O. You will present in class and take questions.

Module Learning Outcomes assessed (exact wording)

This CA assesses the following **module learning outcomes**:

- **MLO1:** Demonstrate an understanding of, and an ability to apply programming features that support Abstract Data Types and the implementation of common data structures and algorithms.
- **MLO2:** Demonstrate an ability to implement and test a variety of algorithms and data structures.
- **MLO3:** Explain the relative advantages and disadvantages of a variety of algorithms and data structures.
- **MLO4:** Select and use appropriate algorithms and data structures for the solution of a variety of problems.
- **MLO5:** Measure the efficiency of different algorithms and data structures.

Topic (choose **one**)

Data Structures: Binary Search Trees; General Trees; Graphs; Tree Maps; Bit Vector.

Algorithm Paradigms: Dynamic Programming; Divide & Conquer; Greedy; Backtracking; Randomized.

Depth over breadth—pick one topic and develop it thoroughly.

Required outputs (**all mandatory**)

1. Screencast (compulsory)

- 6–8 minutes; clear audio. Show code & visuals. Provide **captions or a transcript**.
- **No screencast ⇒ the assignment will not be graded.**

2. Git Repository (compulsory)

- Use Git from the start; push to a **private remote** and **invite the lecturer**.
- Include a minimal build (CMake or VS solution) and a tag for the final submission (e.g., **v1.0-submission**).
- **No repository ⇒ the assignment will not be graded.**

3. Markdown Documentation (compulsory)

- Root **README.md** with a **Table of Contents** linking the following Markdown files:
 - **overview.md** — problem framing, use-cases, selection rationale
 - **design.md** — DS/algorithm explanation with diagrams
 - **code.md** — key C++ snippets; how to build/run; sample inputs

- `analysis.md` — Big-O claims **and** measured results (time/space)
- `contrib.md` — who did what; short reflections
- `references.md` — sources/attributions

◦ Documents must render correctly on the repo. (*Slides alone are insufficient; narration is required.*)

4. In-class Presentation & Q&A

- **6 minutes** presentation + **4 minutes** Q&A. **Both members speak** and may be asked about any aspect.

Scope checklist (what to cover)

- **When to use it:** 2–3 realistic use-cases.
- **How it works:** Step-through with diagrams and **C++** examples.
- **Operations/Steps:**
 - If DS: insert/search/remove (and relevant variations).
 - If Algorithm Paradigm: 1–2 representative problems (e.g., Knapsack, N-Queens, Flood-Fill).
- **Performance:** Big-O table **plus** measured timings/memory; discuss inputs that help/hurt.
- **Trade-offs & alternatives:** Compare against at least one other DS/paradigm.

Marking rubric (100% → scaled to 30%)

- **Clarity & pedagogy — 20%** (explanations, structure, captions/transcript)
- **Technical correctness — 25%** (accurate DS/algorithm; correct complexity; runnable code)
- **Depth & insight — 20%** (non-trivial example(s); meaningful trade-offs; real use-cases)
- **Code quality — 5%** (idiomatic C++; build/run instructions; small tests)
- **Presentation & Q&A — 20%** (timing, delivery, professionalism; confident answers)
- **Teamwork evidence — 10%** (`contrib.md` shows balanced, credible contributions)

Submission & academic integrity

- **Submit via Moodle:** repository link, screencast (file or link), and any required attachments.
- Work must be **entirely your own within the pair**. Do **not** share pseudo-code or source outside your team.
- **Generative AI tools are forbidden for this assignment.** Violations earn **zero** for all involved and are reported to the HoD.

Late policy (institute)

- Late work is accepted **only** with appropriate medical documentation **within 10 working days** of the deadline.
- The standard institute policy for late submissions will apply.