ALGORITHM & DATA STRUCTURES

Final Project

Introduction

Welcome to the final project!

The final project aims at applying the concepts you have acquired during course to a real-world problem.

You will identify a **real-world problem,** determine and justify the **choice of one or more data structures**, **implement** a solution, **analyze your solution** in term of complexity, and finally **document the process** thoroughly.

Projects will be performed in teams. A team consists of 3 students. Good luck everyone!

Agenda

Proposal Submission
*Teams will submit a one-page proposal outlining the problem, ADT(s) to be used, and the planned implementation strategy by **11th December.***

Project Defense and Final Submission
*Teams will deliver a final report, present their project, and defend their solution. Deliverables include the report, code, and presentation. Presentation date: **26th December***

Deliverables

Date	DELIVERABLE
By **11th December**	Proposal Document - *see template*
By **26th December**	Final Report - *see template* Presentation slides Source code

Proposal Document *Template*

The proposal should briefly outline **your problem, plan**, and **approach**. Below is the template with examples to guide you:

Team Members:

Sea Huyty, Phay Someth, Vorn Naratheany

Project Title:

Student Management System

Problem Statement:

Describe the problem you aim to solve and its significance.

Educational institutions face difficulties managing student records efficiently. Outdated systems are prone to errors and delays, making it challenging to retrieve or update student information. The system can improve administrative efficiency and accuracy.

Expected Outcomes:

Summarize what you aim to achieve.

Develop a Student Management System that:

- Enables CRUD operations for student records.
- Stores and retrieves data using files.
- Provides quick search and update functionalities
- Generates reports summarizing student details and statistics.

Proposed ADT(s):

Identify and justify the choice of ADTs.

Binary Search Tree: for storing student's information dynamically and searching efficiently

Implementation Plan:

Outline how you will implement the solution.

File I/O: Implement functions to save and load student records from files to ensure data persistence.

Data Structures: Use a BST to store student records, with each node containing student details (ID, Name, Class, Level)

CRUD Operations:

- **Create:** Insert new student records into the BST.
- **Read:** Search and display student records by ID, Name, Class, or Level.
- **Update:** Modify existing student records in the BST.
- **Delete:** Remove student records from the BST.

Reports: Generate summaries and filtered views of student information, such as lists of students in a particular class or level.

Testing: Validate system functionality and performance through comprehensive testing.