# Mid-Semester Laboratory Project (Graded)

## Timeline

**Due:** 28 Jul by 23:59

**Points:** 10

**Submitting:** a file upload

**File types:** pdf, doc, and docx

**Allowed attempts:** 1

## Instructions:

* Work in a group with a minimum of 3 members and a maximum of 10 members.
* This project must be implemented using the C programming language.
* The group leader is responsible for booking an office hour during Weeks 8 to 12 for a group presentation of the project to the lecturer. The group leader should share the booking link with all group members to ensure their participation. The group project is valued at 5 marks, and each individual contribution is also valued at 5 marks.
* This assessment will contribute to your Continuous Assessment (CA) and impact your final score. Ensure timely submission on or before the deadline.
* Note that any submission after the deadline will result in a deduction of points from your final score.
* For any submission guideline clarifications, please contact your success advisor or schedule an office hour with a lecturer.

## **gitProject Title:** Student Record System

Create a simple Student Record System in C that allows users to manage student information. The program should include the following functionality:

### Initial Setup:

* Create a new C project using JetBrains Clion (or your preferred development environment).
* Write a program that displays a welcome message and prompts the user to enter their name. The program should greet the user by their name.

### Data Storage:

* Implement storage for student information, including name, roll number, and marks, using appropriate data types.

### Input and Output:

* + Add a feature to input a student's marks and display whether the student has passed or failed based on a passing threshold (e.g., marks above 40 are considered passing).

### Student Records Management:

* + Implement a loop that allows the user to input information for multiple students until the user decides to exit.
  + Create an array to store information for multiple students, with the ability to add, remove, and modify student records.
  + Implement functions for adding, modifying, and displaying student records. Ensure these functions take appropriate arguments and return relevant information.

### Memory Management:

* + Use dynamic memory allocation for student records and ensure memory is freed when no longer needed.

### File Operations:

* + Implement the ability to save student records to a text file and load student records from a text file, storing information such as name, roll number, and marks.

### Search Functionality:

* + Allow users to search for a student by their roll number and display the relevant information.

### Calculations and Sorting:

* + Calculate and display the average marks for all students.
  + Provide an option to sort student records based on marks in ascending or descending order.