# Assignment 8 and 9: Unit testing and GUI - School Task Management System

Objective:

# The objective of this assignment is to create GUI front end for the existing School Task Management System.

This is the group assignment. The group should meet and decide what part of the assignment is going to be done by each member of the group. The work should be divide based on gradable features. The code that implements the feature must include the comment with name and student number of the group member that implanted the feature.

Ability to work in the group is 20% of your total mark for the assignment. The rest of the mark will be awarded based on proper coding and working of the application relative to the contribution to the solution that will be determined through the peer review.

# **Please observe the General assignment requirements outlined in the document on e-centennial**

## Requirements:

Part A [10%]:

Create a class library with the code that is submitted in the previous part/assignment and test all methods/properties for one class. Each student in the group should pick a different class.

Part B:

The group should use library of a member to implement the GUI front end with.

### Specification for GUI frontend

#### 1. Overview:

The GUI front application is intended to provide users with an intuitive interface for managing tasks, courses, evaluations, and assignments using the underlying library.

It should allow users to create, edit, and delete tasks, courses, evaluations, and assignments.

Users should be able to view their tasks for the current day as well as tasks associated with specific courses.

#### 2. User Interface Design[20%]:

##### Main Window:

Upon launching the application, users should be presented with a main window displaying options to manage tasks, courses, and evaluations.

Include buttons or tabs for easy navigation between different sections (tasks, courses, evaluations).

##### Tasks Section:

Users should be able to view a list of tasks categorized by status (e.g., pending, completed).

Provide options to add, edit, and delete tasks.

Allow users to mark tasks as completed or set due dates for pending tasks.

##### Courses Section:

Users should be able to view a list of courses they are enrolled in.

Provide options to add, edit, and delete courses.

Allow users to view evaluations and assignments associated with each course.

##### Evaluations Section:

Users should be able to view a list of evaluations (assignments, quizzes, tests) associated with their courses.

Provide options to add, edit, and delete evaluations.

Allow users to specify evaluation details such as due dates, weights, and names.

#### 3. Functionality[20%]:

##### Tasks Management:

Users should be able to add new tasks by providing a description and optional due date.

Tasks should be editable, allowing users to modify descriptions and due dates.

Users should be able to mark tasks as completed or delete tasks.

##### Courses Management:

Users should be able to add new courses by providing course details such as name and code.

Courses should be editable, allowing users to modify course details.

Users should be able to view evaluations and assignments associated with each course.

##### Evaluations Management:

Users should be able to add new evaluations (assignments, quizzes, tests) to specific courses.

Evaluations should be editable, allowing users to modify details such as name, due date, and weight.

Users should be able to specify whether an assignment is a group assignment.

#### 4. Integration with the library[10%]:

Utilize your class library to manage tasks, courses, evaluations, and assignments.

Ensure proper error handling and validation to maintain data integrity and prevent exceptions.

Implement functionalities to save and load data using the persistence methods provided by the library.

#### 5. Additional Features [10%]:

##### My Day View:

Include a dedicated section to display tasks scheduled for the current day (utilize MyDay functionality from the library).

##### Data Export/Import:

Provide options to export tasks, courses, and evaluations data to external files (e.g., JSON format) and import data from existing files.

#### 6. User Experience (UX)[10%]:

Design the application with a user-friendly interface, ensuring ease of navigation and intuitive controls.

Use clear and descriptive labels to guide users through different functionalities.

Provide feedback and notifications to users for successful operations or errors.

Ensure responsiveness and smooth interaction to enhance the overall user experience.

## Submission Guidelines:

Each student should submit the solution as compressed solution folder in Lab 89 drop box by the deadline in the drop box.