GROUP ASSIGNMENT 2 BRIEFING (10%)

Title: "JavaFX GUI for Task Management System"

Deadline: Week 11, Sunday, 25th May 2025.

Learning Outcome:

- CLO2: Implement object-oriented programming systems to address a variety of computing-related challenges, demonstrating problem-solving skills.

Objective:

This assignment continues from Assignment 1. You are now required to develop a basic GUI application using JavaFX to interact with the task management system that you implemented previously. The goal is to apply object-oriented design to a user-facing interface (this term means the GUI is intended for end-users, not internal use) by reusing and extending your existing `Task`, `TaskValidator`, and `TaskManagerApp` classes.

Instructions:

1. Reuse Assignment 1 Classes
    - Import and utilize your `Task`, `TaskValidator`, and `TaskManagerApp` classes.
    - You may modify these classes if necessary to support GUI interaction.
2. Create a JavaFX-based GUI Application
    - Your application must include:
        - `TextField` for task name, category, due date (as text input), and priority.
        - `Button` for adding a task.
        - `Label` for displaying validation feedback.
        - `ListView` or `TextArea` for displaying the list of added tasks.
        - Appropriate layout containers (e.g., `VBox, HBox, GridPane`) to organize the UI.

- The user will enter task details via input fields. On button click:
  - The input is captured to create a `Task` object.
  - `TaskValidator` is called to validate the task.
  - If valid, the task is added to a list and displayed.
  - If invalid, an error message is shown in the `Label`.

3. Styling (Optional but Encouraged)
   - You may enhance UI appearance using:
     - Font
     - Insets and padding
     - Border and color settings
   - However, focus must remain on functionality and layout.
4. Restrictions
   - Use only the basic JavaFX components listed above.
   - Do not use Scene Builder or CSS styling for this assignment.
   - Do not include advanced GUI features like `ComboBox`, `DatePicker`, or `TableView` at this stage.
5. Code Documentation for Modified Classes
   - You may reuse the `Task`, `TaskValidator`, or other classes from Assignment 1 as-is. However, if you make any modifications to these classes (e.g., to support GUI integration), you must insert comments in the code to explain:
     - What was changed
     - Why the change was necessary (e.g., "Added method to convert task to display string for GUI")
   - Code documentation for modifications is encouraged as part of good programming practice and may contribute to a higher grade.

What Comes Next:

This assignment prepares you for the Group Project, where you will extend this system with additional components, improved UI design, and advanced object-oriented features.

Submission Requirements:

- Each group leader must submit the following files:
  - `GUIApp.java` (Main GUI application file)
  - `Task.java` (Task class implementation).
  - `TaskValidator.java` (Validation logic).
  - `TaskManagerApp.java` (Application class).
  - Any helper classes created.
  - Fillable rubric with group information
  - **IMPORTANT**: If you are submitting all files as an archive, ensure that the file format is **.zip**. No other formats (e.g., .rar, .7z, .tar.gz) will be accepted.
- **If there are issues within the group, each group member must submit a peer evaluation. Marks will be deducted based on the average peer evaluation score**.

Grading Rubric Reference:

This assignment prepares you for the Group Project, where you will extend this system with additional components, improved UI design, and advanced object-oriented features.