CCP6224 Object-Oriented Analysis and Design

Lab Exercise

Total Marks: 25% (will be scaled to 40% mathematically, since lab exercise carry 40% in the syllabus)

Due Date: {8 Feb 2026, 11.59pm} (there will not be any extension of due date)

Instructions:

1. This is a group work (a group of 3 to 5 students – students must have already formed the group by now as it was already announced during lectures several times) and you must only work with among your group members. Working with anyone outside your group is considered cheating and will get zero mark. Copying from any other sources will also get zero mark. If you have trouble finding group members, please inform tutor by 19 Dec 2025. Individual work will get zero mark as this is a group exercise.
2. You will be given zero(0) marks if you do not submit on time. You are given ample time to submit earlier. Do not give excuses like internet problems etc.
3. You need to submit the following:

   a. Part 1 deliverables and

   b. Part 2 deliverables and

   c. Part 3 deliverables

   Submit only one zip folder with the file named StudentID-StudentName.zip. The zip folder should contain all the source code (.java files and any resource files), sequence diagrams, a class diagram, and a presentation video. Submission will be online and will be informed later by your tutor.

4. Use only Java Swing to do the exercise. DO not use any other framework.
5. Follow the submission instruction given by your tutor and do submission before the deadline.

Question:

The Faculty of Computing and Informatics (FCI) organizes a Postgraduate Academic Research Seminar each year. This event acts as a mini conference for master's and PhD students to showcase

their research progress and receive constructive feedback from evaluators. To improve management and transparency, the faculty requires a standalone Java Swing–based **Seminar Management System** that can support the following:

Roles

- ***Student (Presenter)***
  - ✓ Registers for the seminar with research title, abstract, supervisor name, and preferred presentation type (Oral/Poster).
  - ✓ Uploads presentation materials (slides or poster file path).
- ***Evaluator (Panel Member)***
  - ✓ Reviews assigned presentations.
  - ✓ Provides evaluation based on predefined rubrics (e.g., Problem Clarity, Methodology, Results, and Presentation)
  - ✓ Adds comments and marks for each presenter
- ***Coordinator (Faculty Staff)***
  - ✓ Creates and manages seminar sessions (date, venue, session type).
  - ✓ Assigns evaluators and presenters to sessions
  - ✓ Generates seminar schedules and final evaluation reports
  - ✓ Oversees award nomination for Best Oral, Best Poster, and People's Choice.
- ***System Requirements***
  - ✓ User Management with role selector for Student, Evaluator, or Coordinator.
  - ✓ Student Module for registration, preference, and upload of presentation files.
  - ✓ Session Management for creating and managing sessions and time slots.
  - ✓ Evaluation Module for scoring and commenting using rubrics.
  - ✓ Poster Presentation Management with board IDs and criteria.
  - ✓ Award & Ceremony Module for computing winners and generating award agenda.
  - ✓ Reports & Summary with export options and data analytics.

Part 1 Deliverables

UML Class Diagram and Sequence Diagrams showing relationships among classes such as User, Student, Evaluator, Coordinator, Submission, Session, Evaluation, Award, and Report.

Part 2 Deliverables

Java Swing implementation fulfilling all core functionalities with a user-friendly GUI.

Part 3 Deliverables

A 10–15 minutes presentation video by the group. Each member must participate and explain on the following (Class Diagram, Sequence Diagram, Implementation, Design Coherence). Refer to the marking rubric below and your presentation sequence should follow the marking rubric (Class

Diagram followed by Sequence Diagram followed by Implementation followed by Design Coherence). Make sure your diagrams and text used in the video are clear as the marking will be based on the video presentation.

Evaluation Rubric:

Criteria	**Poor (1)**	**Below Avg (2)**	**Average (3)**	**Above Avg (4)**	**Good (5)**
Class Diagram	Design meaningless or mostly wrong	Incomplete or wrong UML	Some UML notations missing or unclear	Mostly correct and meaningful	Complete, correct, and realistic design
Sequence Diagram	Missing or incoherent	Partially mapped	Moderately coherent	Logical and coherent	Fully coherent and detailed
Implementation	Failure or incomplete	Partial functionality	Core modules functional	Most features integrated	All modules fully functional
Design Coherence	UML and code unrelated	Partially consistent	Mostly consistent	Consistent and logical	Fully consistent and future-proof
Presentation	Missing or weak	Limited explanation	Acceptable explanation	Clear and confident	Excellent, professional delivery