Lab 4 + Presentation

Part 1: Group Presentation

Exploring the JUnit Framework

- History of JUnit
- Key features of JUnit
- Advantages, disadvantages, and applicability compared to other testing frameworks
- Demo on basic usage of JUnit (including a demonstration of **Test-Driven Development - TDD**)

Part 2: Individual Practice - Writing Test Scripts with JUnit

- **Each group** selects a software project (from courses such as SWP, PRJ, PRF, Lab211).
- **Each member** of the group finds **5 different functions** in the selected software project to write test scripts using JUnit.
- Team members must choose **distinct functions** to ensure variety in the assignment.
- The selected functions should be **sufficiently complex** to allow meaningful test cases with **high coverage**.

Scope and Complexity Requirements for Selected Functions:

- Each function should contain **at least 3 branching conditions** (if/else, switch, loops, etc.) to ensure effective testing. Functions with logic processing, data validation, list filtering, or score calculation are appropriate choices. Before writing the test script, it is necessary to review the selected function's code, identify processing flows, conditional branches, and key input values.
- Test cases must apply **equivalence partitioning** and **boundary value analysis** to ensure comprehensive testing.
- Ensure the test script achieves **100% code coverage**, meaning all possible execution branches are tested.

Submission Requirements:

- **Group presentation slides** on JUnit (PDF or PowerPoint format).
- **Individual test scripts** (Java .java file) along with a **description of the selected functions and test case design approach** (Word .docx or PDF .pdf).
- **Individual report** including:
    - A brief introduction to the selected project.
    - A description of the tested functions.

- Detailed test cases with explanations of applied testing techniques (e.g., equivalence partitioning, boundary value analysis).
- Test execution results and an assessment of test coverage.