The Saigon International University
Faculty of Engineering & Computer Science

Socialist Republic of Vietnam
Independence – Freedom – Happiness
---oOo---

FINAL EXAM

OBJECT ORIENTED PROGRAMMING with JAVA
(ITE23005 – 3 credits)
FORM: FINAL PROJECT ASSIGNMENT

Exam code: **2421** (Semester 2, Academic year 2024 - 2025)
Time to distribute the exam: 22-Apr-2025
Time to submit the exam: 25-May-2025 23:59
Presentation and report time: From 26-May-2025 to 31-May-2025

NOTE INFORMATION

Exam papers and exam data files are provided on the school's LMS system. The data files include ***1 exam PDF file*** (information about the exam, component scores and relevant information).

Students work in groups and do the exercises on their computers, then compress all the files into a compressed file (rar/zip/7z..) and submit it to LMS. The compressed file has the following format: ***Nhom_MaLop.rar*** (or .zip, .7z…)

The files to be submitted include:

- 1 or more zip file (the source code and the database if any),

- 1 word file, which is a summary report of the content that has been done. This is the most important file of this exercise, presented clearly and easily understood, and has a clear, coherent and purposeful topic. The word file requirements are as follows:

+ Presented as an essay, including an introduction (why choose this topic...), content and conclusion.

+ In addition, there must also be a section on teamwork, division of work and assessment of the level of completion of the work of the group members.

+ This is a presentation of all the content that the group has done according to the requirements of the topic.

+ This word file needs to be checked for plagiarism through the turnitin.com portal that the school supports. In case the school does not have enough money for students to use turnitin, the website https://kiemtratailieu.vn/ can be used instead. The requirement is no more than 40% overlap. If checking for plagiarism with turnitin, the student group needs to contact the teacher to provide an account for the group leader to check.

+ It is allowed to use chatGPT or similar AI tools to support writing, however, when using it, you must clearly understand why chatGPT/AI tool writes like that, explain it and be able to prove that the author clearly understands what ChatGPT is saying. If you use ChatGPT or similar tools without explaining or understanding why it is written like that or not clearly stating what chatGPT provides, you will be deducted points or get 0 points.

+ Requires writing in essay format, using references and citations, all must follow APA standards. (https://apastyle.apa.org/)

- 1 ppt file, which is the content of the report to the teacher during the interview (if any).

- 1 Youtube public link, which is a video recording the reporting process. The video recording the final report must include all parts of the assignment, clearly explain why it was done, what the results were... and the issues the group finds necessary to report to the lecturer. Members must participate in the report in the clip and must see the faces of all group members in the report video. Then upload the video file to youtube in public mode and upload it to the drive to share the public link. (If reporting offline, you do not need to record this clip)

CONTENT OF THE EXAM

Section 1 – Topic of the JAVA Application

Topic: Students work in groups (groups of 1-2 students), assigned 1 dataset from the list below:

1) AdventureWorks2019.zip
2) Advertising-campaigns.zip
3) CoffeeShop.zip
4) CyclisticUser.zip
5) Data_jobs_and_skills.zip
6) ERPSystem.zip
7) Health-Response-Score.zip
8) HotelRevenue.zip
9) InstagramAnalysis.zip
10) InventoryManagement.zip
11) MarketplaceTransactional.zip
12) MentalHealth.zip
13) OlympicGames.zip
14) OnlineBookstore.zip
15) OnlineRetailCustomer.zip
16) OnlineShop2025.zip
17) SpyAgency2024.zip
18) UnderwearData.zip
19) menagerie-db.zip
20) sakila-db.zip
21) airport-db.zip

The group of students needs to identify the goal and objective of the application and propose an application to achieve the goal above. It is necessary to ***determine the scope, limitation and desired goals*** of this analysis, from which to build the application as required.

Objective: Build a **Java desktop application** that allows users (e.g., admin/staff) to **log in**, **manage data**, and **visualize statistical insights** using charts. The system must be connected to a **MySQL database**, apply **Object-Oriented Programming (OOP)** principles, leverage **Data Structures**, and display charts using **JFreeChart**.

Important Note on Evaluation Criteria

This assignment is designed as a comprehensive integration of five core professional competencies in software development. To ensure holistic skill development, all five components must be present and functional in the final application.

Any project missing one or more of the following components will be automatically disqualified from evaluation and will receive a score of zero.

Mandatory Skill Components (5/5 Required for Scoring)

1. *Object-Oriented Programming (OOP):* The application must demonstrate clear use of classes, inheritance, encapsulation, and polymorphism.
2. *Database Integration (MySQL) :* The application must connect to a MySQL database and support CRUD operations (Create, Read, Update, Delete) using SQL.

3. *Data Structures:* The program must utilize appropriate data structures from the Java Collections Framework (e.g., ArrayList, HashMap, TreeMap) for storing, retrieving, and processing data efficiently.
4. *User Authentication and Profile Management:* A login/logout mechanism with secure user session handling and a user profile interface must be implemented.
5. *Data Visualization with Charts (JFreeChart):* The application must include at least one dynamic data visualization (e.g., bar chart, pie chart, line chart) to summarize or analyze data.

By successfully completing this assignment, students demonstrate not only technical proficiency but also the ability to synthesize multiple core skills into a single, professional-grade software product.

Section 2 – Core Functional Requirements

Based on your data and target website / system, you can have some of the phases in the penetration testing. Here are the phases of a penetration test, which are suggested, organized systematically as follows:

1. User Authentication
- Login / Logout system
- Passwords stored securely (hashed) in the database
- Basic user profile (name, email, role)

2. Database Integration (MySQL)
- Users table
- Other tables (at least 5 tables)

3. Other Data Management
- Create, Read, Update, Delete (CRUD) data
- Other derivable data

4. Dashboard (JFreeChart)
- Generate statistical charts like: BarChart, Pie, etc.

5. Object-Oriented Programming
- Classes: User, DatabaseConnection, etc.
- Use inheritance, abstraction, and encapsulation

6. Use of Data Structures
- Use ArrayList, HashMap, TreeMap, etc., to:
    - Cache user session
    - Temporarily store and sort list of data
    - Build class-wise statistics
    - Support search and filter operations

Section 3 – Submission Requirements

Each group must submit a **project package** containing the following **deliverables**, organized in a clean folder structure and clearly named:

1. Project Report (PDF or DOCX)

A structured report in English with the following sections:

a. Introduction
- Project title and objective
- Overview of the system

- Purpose and motivation
- Tools and technologies used (Java, MySQL, JFreeChart, etc.)

b. Group details and work distribution

- Full name and student ID of each group member
- Role of each member (e.g., frontend, backend, database, testing)
- A brief summary of how the team collaborated

c. System features

- Description of main features (login, CRUD, charts, etc.)
- Screenshots of key components and UIs

d. Technical architecture

- Description of the system design (MVC or similar)
- UML class diagram(s)
- Entity-Relationship Diagram (ERD) of the database

e. Use of Java concepts

- Explanation of Object-Oriented Programming in the project
- Data structures used (with justification)

f. Data visualization

- Description of the charts implemented using JFreeChart
- Explanation of how the visualizations support the application

g. Conclusion and recommendations

- What the group learned
- Challenges faced and how they were overcome
- Recommendations for improving the application or future versions

h. References

- Links to libraries used (e.g., JFreeChart)
- Any tutorials, articles, or documentation consulted
- Follow APA or IEEE style (choose one)

2. Source vode folder

- Full Java source code with:
    - Proper file/folder structure
    - Code comments
    - A README.md file explaining how to run the program
    - gitHub links (public access)
- Include:
    - SQL scripts to create and populate the database
    - .jar executable (optional, for demo)

3. Demo video (May be stored in Google Drive or Youtube)

- 3–5 minute screen-recorded video walking through:
    - Login/logout
    - CRUD operations
    - Data visualization (charts)
- Narration or subtitles explaining the steps

4. PPT for presentation

Section 4 – Rubrics for evaluating the final assignment

Criteria	Excellent (Full Points)	Good (Mid-High Points)	Average (Mid-Low Points)	Poor (Low or No Points)	Max Points
1. Login / Logout / Profile System	Secure login/logout with proper session handling and profile management.	Functional login/logout and profile, minor UI or logic issues.	Basic login/logout, weak session or profile logic.	Missing or completely non-functional.	10
2. Database (5 tables above) **CRUD Operations**	Full CRUD on data with MySQL; no errors.	CRUD mostly functional, few bugs or missing checks.	Limited CRUD (e.g., only Read & Insert), unstable connection.	No CRUD or database not integrated.	20
3. Object-Oriented Design (OOP)	Strong class structure, well-organized OOP with abstraction, encapsulation, and inheritance.	Good OOP use, though some class designs are redundant or coupled.	Minimal OOP, mostly procedural with few class abstractions.	Poor or no OOP principles applied.	20
4. Use of Data Structures	Smart and efficient use of multiple data structures (ArrayList, HashMap, etc.) in logic and storage.	At least two data structures used with moderate relevance.	Only basic structures used; minimal application in solving tasks.	No clear use of Java collections or logic too static.	10
5. JFreeChart Data Visualization	Well-designed dashboard with multiple meaningful and interactive charts.	Charts displayed with useful insights but limited variety or clarity.	One basic chart; data may be static or misrepresented.	No chart or incorrect use of JFreeChart.	20
6. Code Quality & UI	Clean, modular code with comments; professional UI with full functionality.	Mostly clean code; UI is decent but not intuitive.	Messy or hard-to-read code; UI is cluttered or incomplete.	Unreadable code, no comments, and UI is missing or non-functional.	10
7. Documentation & Presentation	Clear README with setup guide, DB schema, class explanation, and screenshots.	Good documentation but missing 1-2 components (e.g., UML or screenshots).	Basic README with little detail or unclear instructions.	No documentation or incomplete.	5
8. Extra Features / Creativity	Innovative features beyond requirement (e.g., search, filter, export to Excel/PDF).	One small extra or UX improvement.	No extra features but minimum requirements met.	Not attempted or does not meet minimum.	5

Oral examination / Live demo guidance: During the review session, ask each group to:

- Walk through their application's workflow
- Explain their code logic and design choices
- Justify data structure and chart selections
- Demonstrate CRUD operations and dashboard
- Reflect on their learning experience and team collaboration

Grading scale

Score Range	Grade	Interpretation
90 – 100 pts	A (Excellent)	Highly proficient and innovative
75 – 89 pts	B (Good)	Solid work with minor improvements needed
60 – 74 pts	C (Average)	Meets expectations, but lacks depth
40 – 59 pts	D (Poor)	Incomplete, limited understanding
Below 40 pts	F (Fail)	Failed to meet core requirements

This table format provides a clear and detailed rubric for evaluating the assignment, ensuring objectivity and transparency in grading.

(End)