PROJECT

Object Oriented Programing with Java

Semester: Fall 2025

Worked By: Anxhela Likcani and Eliora Qendro

University of New York Tirana

Faculty of Engineering and Architecture (Software Engineering)

Instructor: Elton Ballhysa

TIRANA 2026

Report: Student Registry System

1.Project Context and Overview

This project was developed as part of the OOP with Java course at University of New York Tirana. The goal of the assignment is to design and implement a command-line based student registry system using Java, applying object-oriented programming principles and file handling.

2.Problem Statement

The student registry system is responsible for managing students, courses and grades. All data is stored in text files and loaded into memory when the program runs. The system allows users to interact through a Command Line Interface (CLI) to load data, search for records, query data, add new entries, and generate reports.

3. Functional Requirements

The application supports the following commands:

- load: Loads students, courses, and grades from text files

- find: Finds a specific student, course, or grade by unique key

- query: Searches records using filtering conditions

- add: Adds new students, courses, or grades

- report: Generates reports (top students, top courses, transcript)

- quit: Terminates the program gracefully

4. Non-Functional Requirements

The application is designed to handle large datasets efficiently, respecting memory constraints and ensuring fast execution time. All operations complete within acceptable time limits as specified in the assignment.

5. Technical Design & Implementation

The system follows object-oriented design principles. Each entity is represented as a separate class:

- Student: Stores student information and GPA calculation

- Course: Stores course details and determines course level

- Grade: Stores evaluation information and computes letter grades

A central Registry class manages all data and operations. File handling is implemented using Java I/O to read and write text files.

6. Data Structures & Algorithms

Collections such as HashMap and ArrayList are used to store and retrieve records efficiently. HashMaps allow fast lookup by unique identifiers, while lists are used for sorting and report generation.

7. Error Handling

The program validates all inputs according to the rules defined in the assignment. Invalid commands, missing parameters, duplicate entries, and incorrect data formats are handled gracefully with clear error messages displayed to the user.

8. Verification & Testing

The system was tested using multiple datasets to verify correct functionality. All commands were tested with valid and invalid inputs to ensure robustness.

9. Achievements & Challenges

This project strengthened understanding of object-oriented programming, file handling, and command-line application design. The main challenges included managing file persistence and implementing accurate GPA calculations.

10. Future Improvements

Future improvements could include integrating a database, adding a graphical user interface, or supporting additional reporting features.

11. Conclusion

The student registry system successfully meets the functional and non-functional requirements of the assignment. It demonstrates effective use of Java, object-oriented design, and file processing.