# **Programming Assignment Unit 7**

Computer Science, University of the People

CS 1101-01 Programming Fundamentals - CS 1102-01 - AY2024-T2

Instructor, Noman Shihadeh

January 03, 2023

## Student Course Management System + GUI

For this assignment, we were asked to write a program that manages a collection of students and courses. We also need to allow the students to register for a course and allow setting a student’s grade for that course. We need to build the solution using OOP and encapsulation principles. Also unlike previous assignments, this application will be wrapped and displayed to the user using a visual GUI, specifically using JavaFX. The system will have the following controls and abilities:

1. Students:
   1. Allow adding, editing and deleting students
2. Course:
   1. Allow adding, editing and deleting courses
3. Grade and Enrollments
   1. Allow enrolling students to courses and setting a grade for that combination

Source Code:

The source code for this assignment will be found alongside this document file as .java files in folders in the following structure:

1. “controllers” – this folder contains the controllers that will manage the data for each of the 3 sources, Students, Courses and Grades.
2. “models” – this folder contains the model classes that represent the objects in our application Students, Courses and Grades.
3. “gui” – this folder contains the panels that will represent each of our models and controllers. Allowing adding, editing and deleting data for each. This folder also contains the “MainFrame” which is the entry point of our application and build all the components and stitches them together.
4. “utils” – this folder contains a few utility classes that will be used in different parts of the code.

The main parts of the code:

1. Controllers: Located in the controller's folder, these classes manage the application's data logic. They include:
   1. BaseController: An abstract class containing shared properties and methods for managing lists of students, courses, and grades.
   2. StudentController: Manages student-related operations such as adding, updating, and deleting students.
   3. CourseController: Handles course-related tasks including adding, updating, and removing courses.
   4. GradeController: Responsible for grade and enrollment functionalities such as registering students to courses, setting grades, and removing grades.
2. Models: Found in the models folder, these classes represent the application's data structures:
   1. BaseEntity: An abstract class providing a common ID property for derived models.
   2. Student: Represents student data with properties for name and email.
   3. Course: Contains course information including the course name.
   4. Grade: Represents a student's grade in a course, including methods for calculating letter grades.
3. Graphical User Interface (GUI): Stored in the gui folder, these classes build the user interface:
   1. BasePanel<T>: An abstract generic class providing common functionality for different types of panels in the GUI.
   2. StudentPanel, CoursePanel, GradePanel: Specific panels for managing students, courses, and grades, extending BasePanel.
   3. MainFrame: The main application window that sets up the primary stage and tabs for each panel.
4. Utilities: Located in the utils folder, these classes offer additional functionality:
   1. ValidationUtils: Contains methods for input validation, such as checking for non-empty strings and valid email formats.
   2. GradeUtils: Provides functionality to convert numeric grades to letter grades.

Each part of the code plays a critical role in the application. Controllers manage the business logic, models represent the data, GUI classes create the visual interface, and utility classes offer additional functionalities to support various operations within the application.

**Output (screenshots)**

Screen 1 – manages the students data and list

A screenshot of a computer

Description automatically generated

Screen 2 – manages the courses data and list

A screenshot of a computer

Description automatically generated

Screen 3 – manages the student course registration and grade managment

A screenshot of a computer

Description automatically generated

**Conclusion**

The Student Course Management System, as outlined in this document, represents a comprehensive solution for managing students, courses, and grades in an educational setting. Through the application of object-oriented programming principles and the utilization of JavaFX for the graphical user interface, the system offers a user-friendly and interactive platform for administrative tasks.

The system is structured into distinct but interrelated components, each playing a pivotal role in the overall functionality. The controllers (StudentController, CourseController, GradeController) serve as the backbone of the application, handling the business logic and data manipulation. The models (Student, Course, Grade) encapsulate the core data, while the GUI classes (StudentPanel, CoursePanel, GradePanel, MainFrame) provide the visual representation and user interaction capabilities.

Utility classes (ValidationUtils, GradeUtils) enhance the system's robustness by offering validation and grading functionalities, ensuring data integrity and consistency. The inclusion of error handling, user input validation, and a straightforward yet functional user interface demonstrates a focus on reliability and user experience.

As technology and educational needs evolve, this system can be expanded and refined. Potential enhancements could include integrating more advanced features, optimizing performance, and expanding its scope to cover more aspects of educational management.

In conclusion, this document and the accompanying source code provide a detailed overview of a versatile and efficient Student Course Management System. Its modular design, combined with an intuitive user interface, makes it a valuable tool for educational institutions, simplifying and streamlining the management of student records, course offerings, and grading.

## References

Java Language and Virtual Machine Specifications

* <https://docs.oracle.com/javase/specs/index.html>

Introduction to Programming Using Java - Version 9.0, JavaFX Edition

* <https://math.hws.edu/javanotes/>

JavaFX Documentation Project

* <https://fxdocs.github.io/docs/html5/>

CS 1102-01 - AY2024-T2 - Learning Guide Unit 7

* <https://my.uopeople.edu/mod/book/view.php?id=391999>

**General Disclaimer**: All names and terms that are trademarks are used here for educational and illustrative purposes only, without any intent of infringement. All rights to these trademarks are reserved by their respective owners.