# **DEVELOPING AN E-LEARNING PLATFORM: AN OBJECT-ORIENTED APPROACH**

# A Case Study Presented to Mr. Joseph Lorilla of University of Southern Mindanao

In Partial Fulfillment of the Requirements in the Object-Oriented Programming Course

By:

**Acosta, Keandra Nathalie**

**Idsla, Hasmin**

**Sali, Baililian**

November 2024

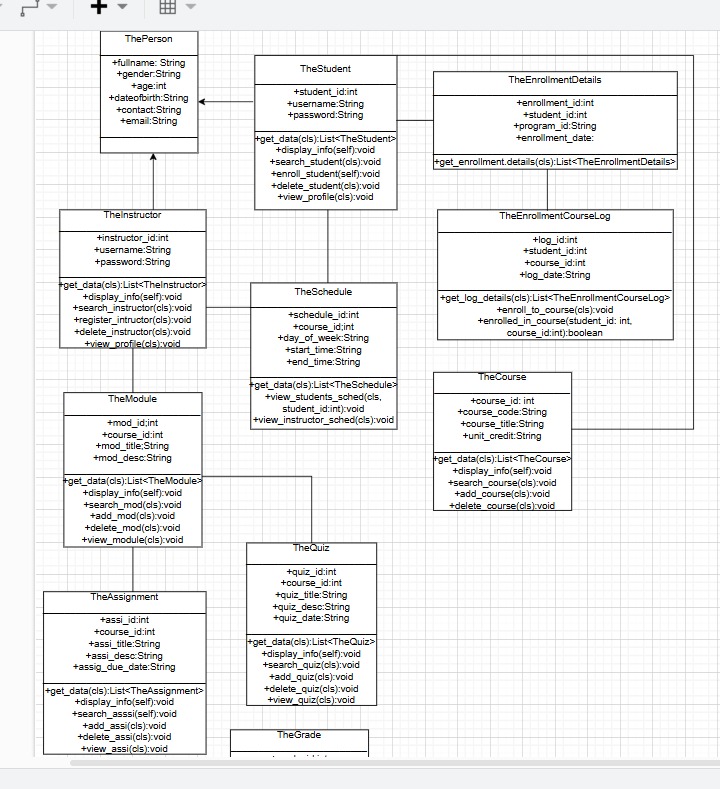
**I. INTRODUCTION**

This case study examines the development of an E-Learning Platform designed to efficiently manage courses, students, instructors, and their interactions. The platform enables students to self-enroll in courses, instructors to manage assignments and grades, and administrators to oversee platform operations, ensuring a cohesive learning experience. Key features include user management, course scheduling, and enrollment tracking, supported by specialized classes such as Course, Enrollment, Assignment, and Grade for streamlined functionality.

The platform employs Object-Oriented Programming (OOP) principles like encapsulation, inheritance, and polymorphism to ensure a modular and secure structure. A base class, Person, underpins shared functionalities, with subclasses for Student and Instructor to address specific needs. Encapsulation safeguards data integrity, while static methods enhance system efficiency. The integration of a Microsoft SQL Server (MSSQL) database ensures secure, scalable, and real-time management of users, courses, and transactions, bolstering reliability and performance.

This case study highlights the design and implementation of the E-Learning Platform, showcasing how OOP techniques and robust database management create a user-friendly and scalable educational solution tailored to modern e-learning needs.

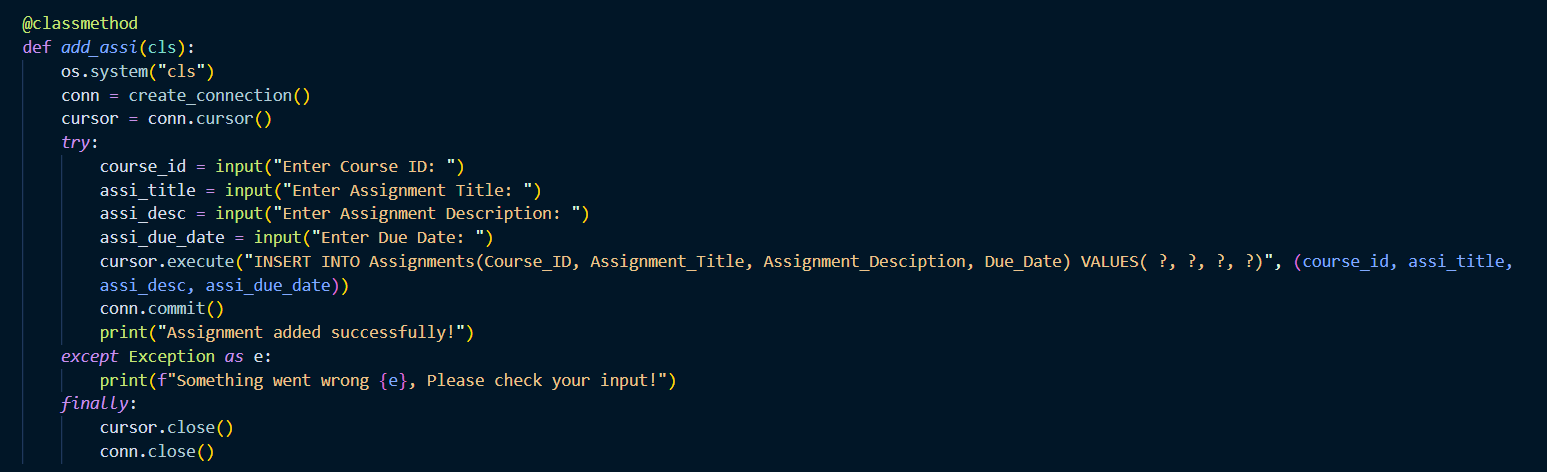
**II. CLASS DIAGRAM**

****

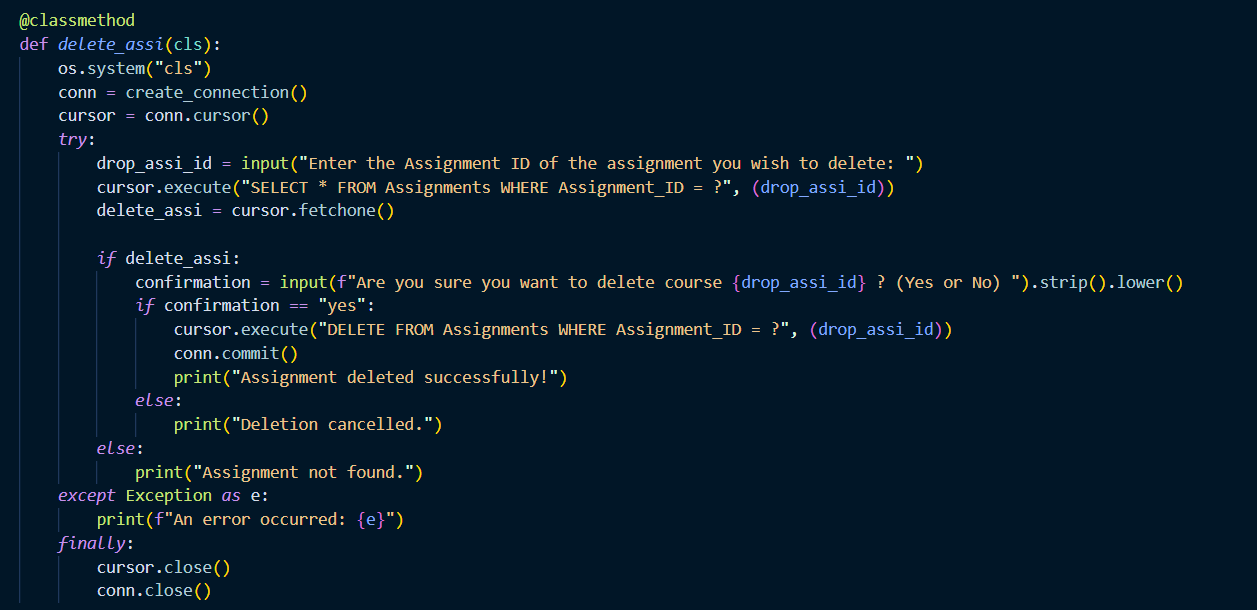
* The diagram shows the class diagram for a system that manages student enrollment, courses, assignments, quizzes, and grades.
* **III. CODE SNIPPETS**

****

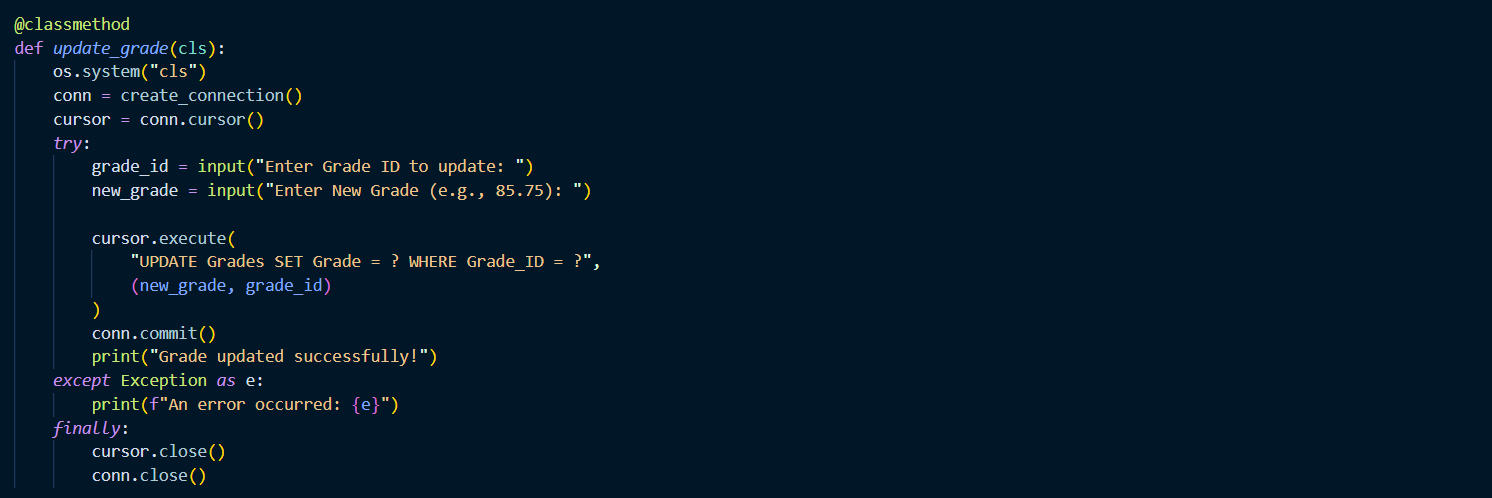
* This function retrieves all data from a specific table in the database, converts each entry into an object, and returns a list of these objects. This approach is consistent across all classes with data stored in the database.



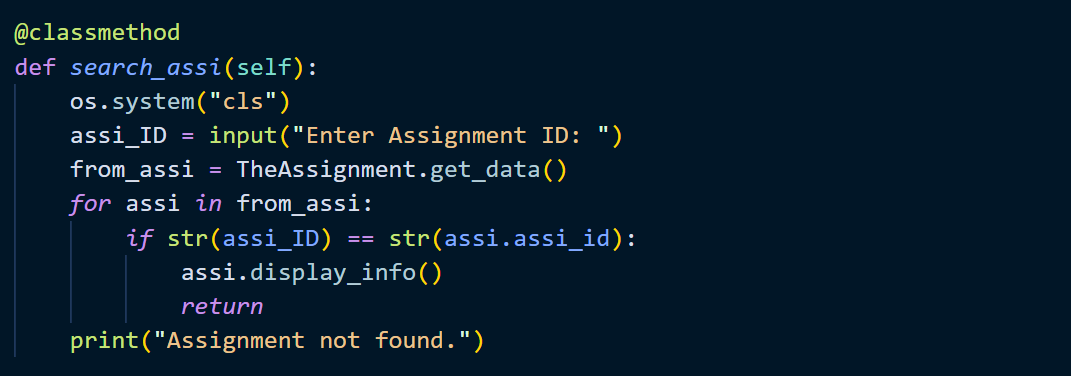
* This method is responsible for adding new assignments to the database. It behaves similarly to other classes that offer this feature.



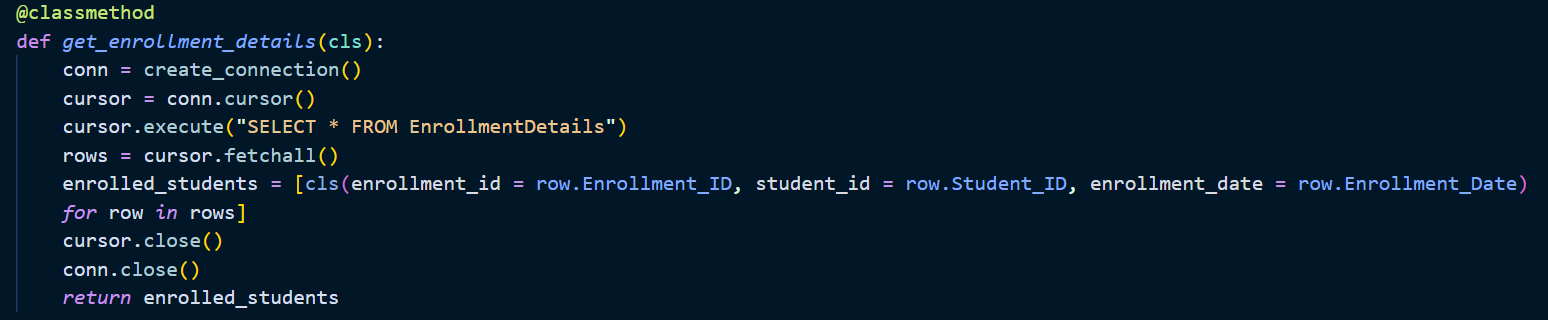
* This method deletes an existing assignment from the database. It behaves similarly to other classes that offer this feature.



* This method updates a grade in the database.



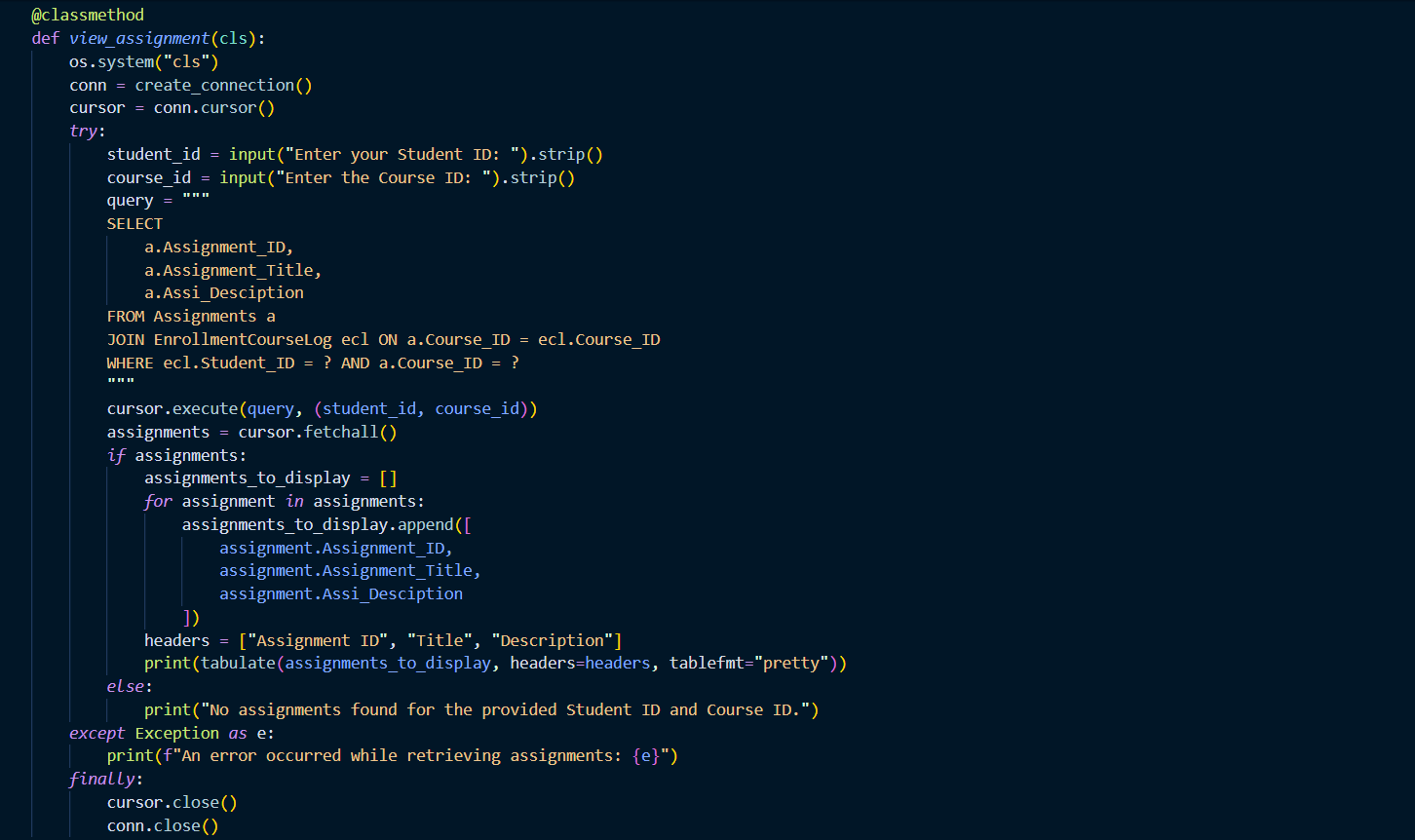
* This method searches for an assignment based on its ID. It prompts the user to enter an assignment ID and then iterates through a list of assignments to find a match. It behaves similarly to other classes that offer this feature.



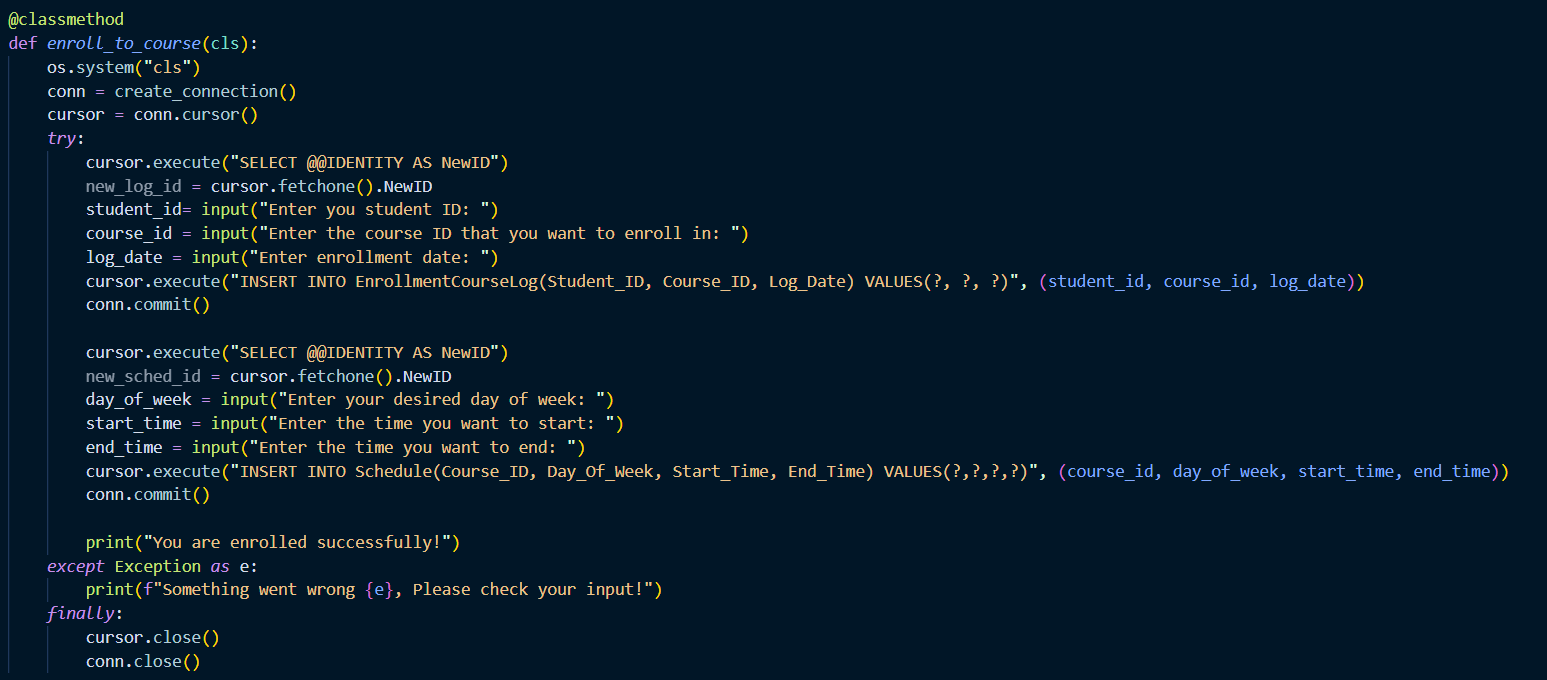
* This function fetches enrollment details from a database. It connects to the database, retrieves all data from a particular table, and stores it in a list. It behaves similarly to other classes that offer this feature.



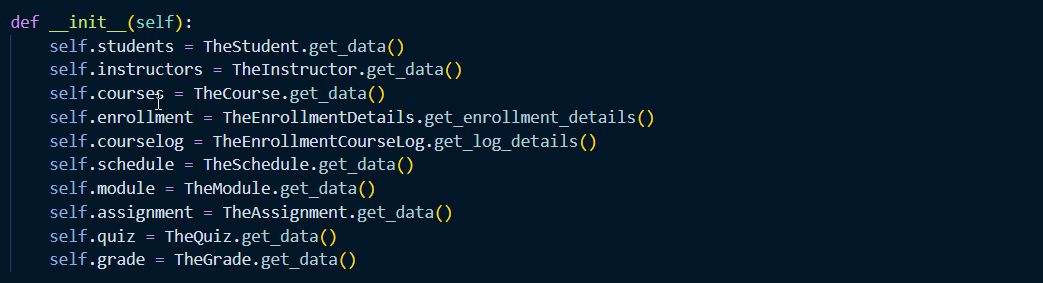
* The function displays information directly to the console. It behaves similarly to other classes that offer this feature.



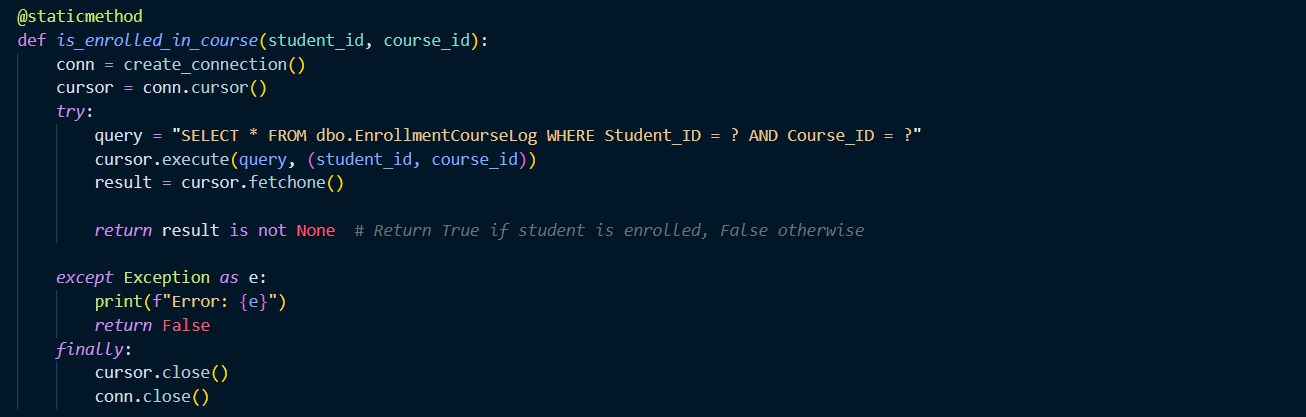
* This function retrieves assignments for a certain student and course. It prompts the user for their student ID and course ID before executing an SQL query to fetch assignments from a database. It behaves similarly to other classes that offer this feature.



* This function allows a student to enroll in a course. It behaves similarly to other classes that offer this feature.



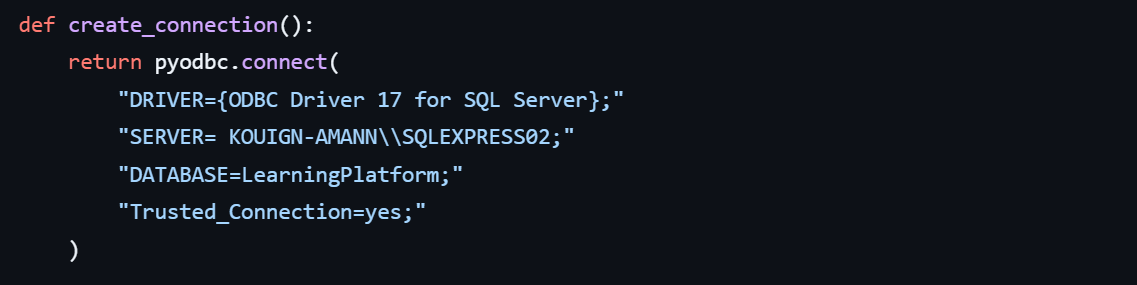
* This function initializes an object by retrieving data from different sources. It behaves similarly to other classes that offer this feature.



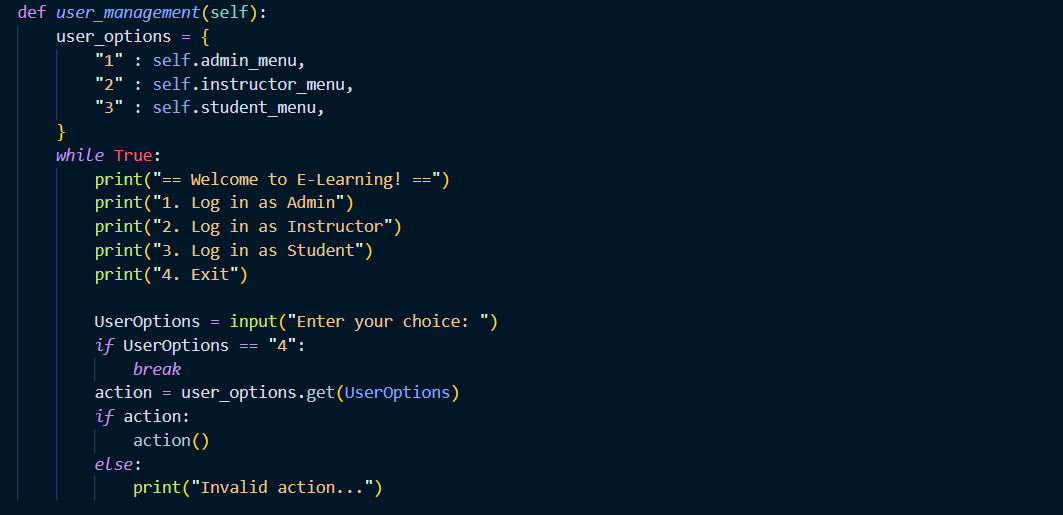
* This function checks if a student is enrolled in a specific course. After receiving the student ID and course ID as input, it runs an SQL query to look for a record in the database that matches.



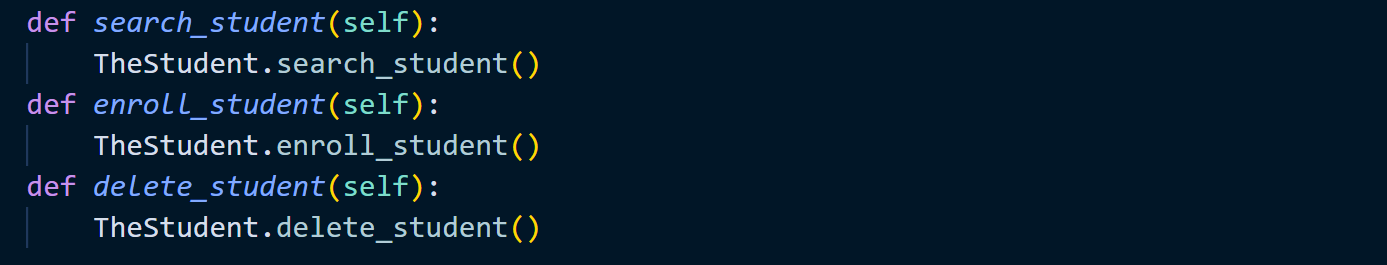
* This method displays full student and enrollment information in a structured table format. It behaves similarly to other classes that offer this feature.



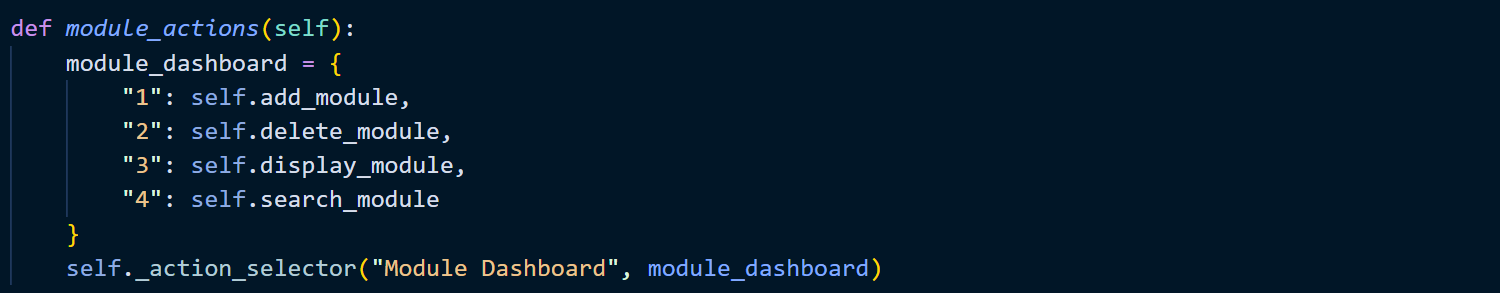
* These are the credentials required to connect to the database. This information will be used whenever a connection to the database is established.



* This function manages user interactions in an E-learning system. It displays a menu with login options. It prompts the user to enter their choice and then executes the corresponding action. This means that when a user picks a number associated with a function, that function will be triggered. It behaves similarly to other classes that offer this feature.



* Calling the functions from here to the dictionary. It behaves similarly to other classes that offer this feature.



* The function is responsible for presenting the menu to the user, getting their input, and then calling the correct method from the dictionary. It behaves similarly to other classes that offer this feature.