

Designing an Object-Oriented Stylized University System (1)

Introduction

In this exercise, you will develop a simple university system comprising courses, students, and teachers. You will practice using classes, properties, and methods in Python to model these entities and their interactions.

Task Overview

Your goal is to create three classes:

1. **Course** - Represents a university course.
2. **Student** - Represents a student enrolled in courses.
3. **Teacher** - Represents a teacher teaching courses.

Each class should have specific attributes and methods as described in the following sections.

The Course Class

- Initialize the **Course** class with a name.
- Generate a unique identifier (ID) for each course instance.
- Maintain a list of student IDs for students enrolled in the course.
- Store the ID of the teacher assigned to the course, if any.
- Implement a method to add a student's ID to the course.
- Allow defining a teacher for the course through a method.
- Provide a readable representation for course instances.

The Student Class

- Initialize the **Student** class with a name and surname.
- Generate a unique student ID for each student instance.
- Keep track of the IDs of courses in which the student is enrolled.
- Implement a method for enrolling in a course, ensuring no duplicate enrollments.
- Provide a readable representation for student instances.

The Teacher Class

- Initialize the **Teacher** class with a name and surname.
- Generate a unique teacher ID for each teacher instance.
- Keep track of the IDs of courses taught by the teacher.
- Implement a method for teaching a course, ensuring no duplicate course assignments.
- Provide a readable representation for teacher instances.

Additional Instructions

- Use the `@property` decorator to create read-only access to the unique IDs in each class.
- Try your code with a simplified example.

Conclusion

By completing this exercise, you will gain hands-on experience with defining classes, managing state, and implementing interactions between objects in Python. Focus on the principles of encapsulation and the use of properties to control access to object attributes.