Student Grade Management System

Description:

The Student Grade Management System is a program designed to manage student and course information, record grades, calculate GPAs, and generate grade reports. It provides users with a user-friendly interface to perform various operations, such as adding students, adding courses, recording grades, and generating reports.

Classes:

1.Person: This class serves as a base to store common attributes like ID and name for both students and courses.

Methods:

getId(): Retrieves the ID of the person.

getName(): Retrieves the name of the person.

serialize(): Converts the person object into a string.

deserialize(): Converts a string into a person object.

2. Course: This class inherits from Person and stores information about courses.

Inherits from: Person

Attributes:

courseId: ID of the course.

courseName: Name of the course.

Methods:

serialize(): Converts the course object into a string.

deserialize(): Converts a string into a course object.

3. Grade: This class stores grade information for each course.

Attributes:

courseId: ID of the course.

score: Grade obtained in the course.

Methods:

serialize(): Converts the grade object into a string.

deserialize(): Converts a string into a grade object.

4. Student: This class inherits from Person and stores student information along with their grades.

Inherits from: Person

Attributes:

grades: Vector to store grades for each course.

Methods:

addGrade(): Adds a grade for a course.

calculateGPA(): Calculates the GPA of the student.

serialize(): Converts the student object into a string.

deserialize(): Converts a string into a student object.

Functions:

1. saveToFile(): Saves student and course data to files.
2. loadFromFile(): Loads student and course data from files.
3. addStudent(): Adds a new student to the system.
4. addCourse(): Adds a new course to the system.
5. recordGrade(): Records a grade for a student in a course.
6. generateGradeReport(): Generates a report displaying student GPAs.

Data Persistence:

Student and course data is stored in "students.txt" and "courses.txt" files to ensure that it persists between program executions.

At program start, data is loaded from these files, and any modifications made during the program execution are saved back to the files.

User Interface:

The program provides a simple command-line interface for users to interact with.

Users can choose from options such as adding students, adding courses, recording grades, calculating GPAs, and generating grade reports.

Error Handling:

The program includes basic error handling for file operations and user input validation.

Note: This documentation offers a general overview of the code's structure and functionality. For more detailed information, refer to the code comments and function definitions.