Title: Comprehensive Virtual Classroom Management System Utilizing Operator Overloading and Additional Functionalities

Objective: Develop a comprehensive virtual classroom management system that allows teachers to efficiently manage students, assignments, class activities, communication, and performance tracking. This project will showcase your understanding of object-oriented programming, particularly operator overloading, as well as your ability to design and implement various features to create a complete and functional system.

Project Description: Design and implement a comprehensive virtual classroom management system that enables teachers to perform various tasks such as adding and removing students, creating and assigning tasks, managing class activities, communicating with students, and tracking student performance. Utilize operator overloading in the implementation of specific classes to enhance the functionality and ease of use of the system.

Key Features:

1. Student Management: Implement a class called "Student" to store student-related information such as name, ID, email, and current grades. Overload the '+' and '-' operators to add and remove students from a list of enrolled students easily.
2. Assignment Management: Implement a class called "Assignment" to store assignment-related information such as title, description, due date, maximum points, and attached files. Overload the '<' and '>' operators to allow easy comparison of assignments based on their due dates.
3. Class Activities: Implement a class called "ClassActivity" to manage various class activities such as quizzes, discussions, and group projects. Overload the '==' and '!=' operators to enable easy comparison of activities based on their type and scheduled date.
4. Communication: Implement a class called "Communication" to facilitate communication between teachers and students. This class should enable teachers to send announcements, individual messages, and group messages. Students should be able to respond to messages and participate in group discussions.
5. Performance Tracking: Implement a class called "PerformanceTracker" to monitor and analyze student performance over time. This class should allow teachers to track grades for individual assignments, calculate overall grades, and identify areas where students may need additional support.
6. Scheduling and Calendar: Implement a class called "Calendar" to manage important dates, such as assignment due dates, exam dates, and class activity schedules. Overload the '+=' and '-=' operators to add and remove events from the calendar easily.
7. Classroom Management: Implement a class called "VirtualClassroom" that integrates the management of students, assignments, class activities, communication, performance tracking, and scheduling. Overload the '[]' operator to allow easy access to the enrolled students, assignments, activities, messages, and events using their index in the respective lists.
8. User Interface: Design a user-friendly interface for teachers and students to interact with the virtual classroom management system efficiently. This interface should enable teachers to perform all the necessary actions to manage their virtual classroom, and students should have access to their assignments, messages, and class activities.
9. Data Storage: Save and load data from an external file or database to maintain the state of the virtual classroom across different sessions.
10. User Authentication: Implement a secure user authentication system to ensure that only authorized users can access the virtual classroom management system.

Complete the following project prompt.

Project Prompt: Virtual Classroom System

You have been assigned to create a virtual classroom system that allows teachers to manage their classes and students. The system should enable teachers to create classes, add students to classes, and manage grades for assignments and exams. The system should also enable students to view their grades and submit assignments.

Student class:

The Student class should represent a student and have the following functions:

add\_grade(string assignment\_name, double grade): This function should add the grade for the specified assignment to the student's record of grades.

view\_grades(): This function should display the grades for all assignments for the student.

submit\_assignment(string assignment\_name): This function should mark the specified assignment as submitted by the student.

Overload the << operator to display the student's name, ID number, and list of submitted assignments.

Class class:

The Class class should represent a class and have the following functions:

add\_student(Student student): This function should add student to the list of students in the class.

remove\_student(int ID): This function should remove the student with the specified ID from the list of students in the class.

add\_assignment(string assignment\_name): This function should add the specified assignment to the list of assignments for the class.

add\_grade(int ID, string assignment\_name, double grade): This function should add the grade for the specified assignment to the student with the specified ID.

view\_students(): This function should display the names, ID numbers, and list of submitted assignments for all the students in the class.

Overload the << operator to display the class name and list of assignments for the class.

Teacher class:

The Teacher class should represent a teacher and have the following functions:

create\_class(string class\_name): This function should create a new class with the specified name and add it to the teacher's list of classes.

remove\_class(string class\_name): This function should remove the class with the specified name from the teacher's list of classes.

add\_student\_to\_class(int ID, string class\_name): This function should add the student with the specified ID to the class with the specified name.

add\_assignment\_to\_class(string assignment\_name, string class\_name): This function should add the specified assignment to the class with the specified name.

add\_grade(int ID, string assignment\_name, string class\_name, double grade): This function should add the grade for the specified assignment to the student with the specified ID in the class with the specified name.

view\_classes(): This function should display the names and list of assignments for all the classes the teacher is managing.

VirtualClassroomSystem class:

The VirtualClassroomSystem class should represent the virtual classroom system and have the following functions:

login(string username, string password): This function should log in the user with the specified username and password. If the username and password match a teacher's credentials, the function should return a Teacher object. If the username and password match a student's credentials, the function should return a Student object. If the username and password do not match any user's credentials, the function should return an error message.

create\_teacher(string username, string password): This function should create a new teacher account with the specified username and password and add it to the system's list of teachers.

create\_student(string username, string password): This function should create a new student account with the specified username and password and add it to the system's list of students.

The VirtualClassroomSystem class should maintain a list of all the teachers and students in the system. When a teacher or student is created, they should be added to the appropriate list.

The system should store all data in a database to allow for