Object Oriented Programming Summer-2023-Project

Total Points: 100
INSTRUCTIONS

- Plagiarism in course project will result in F grade in the course
- This is not a group project and each person will be working on the project individually.
- Make sure you submit your project before the submission time. Late submissions won't be accepted even if they are late by just one minute.
- You can earn bonus marks by implementing extra features in the project.
- Use good programming practices (well commented and indented code; meaningful variable names, readable code etc.).
- Combine all your work in one folder and compress it into a zip file. The folder must contain .cpp .h files (no binaries, no exe files etc.). E.g. i20xxxx_project.zip
- Submit the solutions via google classroom. Submissions via email will not be accepted.

Case Study: Advanced Console-based Classroom Management System

Introduction:
Alex, a student, embarks on developing a Console-based Classroom Management System as a personal project. The system aims to replicate the functionalities of popular virtual classroom platforms like Google Classroom using the C++ programming language and object-oriented programming (OOP) concepts. Alex intends to create a user-friendly, efficient, and comprehensive system to handle user management, class organization, assignment management, submission and grading, gradebook management, file handling, notification delivery, user interface interactions, data persistence, and exception handling.

User Management:
In the User Management system, Alex creates classes to represent users with attributes such as name, email, role, unique ID, password, and contact information. The implementation uses OOP concepts like inheritance to handle different user roles (teachers and students) and polymorphism to handle authentication and user registration. File handling techniques are employed to save and retrieve user profiles, ensuring that user data is persisted even when the program is closed. Alex also implements password management features, such as encryption and validation, to enhance security.

Class Management:

The Class Management system allows users to create and organize virtual classrooms. Each class is represented as a class object with attributes like class name, subject, teacher, description, and a roster of enrolled students. Alex uses aggregation and composition relationships to manage the association between classes and users. The system enables users to create, update, and delete classes, and view class details. Enrolled students can be added or removed from class rosters. Class data is stored persistently using file handling techniques, ensuring that information is saved between program runs.

Assignment Management:
Alex develops the Assignment Management system to handle assignments within classes. Each assignment is represented as an assignment object with attributes like title, description, deadline, attached files, and point value. Encapsulation and abstraction ensure data integrity and modularity in the system. The system supports functionalities for creating, updating, and deleting assignments. Validations prevent the creation of assignments with past deadlines. File handling techniques are employed to store and retrieve assignment data, including details and attached files.

Submission and Grading:
The Submission and Grading system allows students to submit assignments, and teachers can review and grade them. Submissions are represented as submission objects with attributes like submission timestamp, attached files, evaluation status, and feedback. OOP concepts like inheritance and polymorphism handle different submission types, such as text or file uploads. Validations are included to prevent late submissions. File handling techniques are used to store and manage submission and grading data, including student submissions, grades, and feedback.

Gradebook:
The Gradebook system provides an overview of student grades and class averages. A gradebook class calculates and updates student grades based on assignment scores and weightage. The gradebook includes attributes such as cumulative grades, class averages, and weightage settings. Encapsulation and abstraction ensure accurate grade calculations and easy retrieval. The system allows for generating grade reports or transcripts for individual students or the entire class. File handling techniques are employed to store and retrieve gradebook data, including individual student grades, class averages, and weightage settings.

File Management:
The File Management system handles files related to classes, assignments, and submissions. A file manager class handles file upload, download, and deletion operations. The system enforces file size limits and appropriate formats. Encapsulation and abstraction ensure modularity, and file handling techniques efficiently store and manage file data, ensuring organized file retrieval and storage.

Notification System:
Alex implements a Notification System to keep users informed about important updates and deadlines. Notifications are generated for new assignments, upcoming deadlines, and graded submissions. Users can customize their notification preferences, and notifications can be displayed in the console. Polymorphism and abstraction handle different notification

types. File handling techniques are used to store notification data, including settings, timestamps, and delivery status.

User Interface:
To provide a user-friendly experience, Alex develops a console-based User Interface (UI). The UI features menus, prompts, and dialogues for easy interaction with the system. Encapsulation and abstraction are used to encapsulate UI components and ensure modularity. Proper error handling and informative prompts guide users through the system's functionalities. User interface preferences, such as window size and layout customization, are saved using file handling techniques.

Data Persistence:
To ensure data longevity and reliability, Alex employs robust file handling techniques for data persistence. User profiles, class information, assignment details, submissions, grades, and notifications are stored in files. This ensures that data is saved and retrieved even when the program is closed or reopened. File reading and writing operations are used to maintain data integrity and facilitate easy data retrieval.

Exception Handling:
To handle runtime errors and exceptional conditions gracefully, Alex incorporates comprehensive exception handling mechanisms. Custom exceptions provide meaningful error messages and handle scenarios like file I/O errors or invalid user input. Exception handling prevents program crashes and provides error logs for troubleshooting, ensuring a stable system.

Conclusion:
In this case study, Alex, a student, successfully developed an Advanced Console-based Classroom Management System using C++ programming language and OOP concepts. The system encompasses essential functionalities for user management, class organization, assignment handling, submission and grading, gradebook management, file handling, notification delivery, and user interface interactions. Alex's proficient use of inheritance, polymorphism, encapsulation, abstraction, and file handling techniques demonstrates their understanding of key OOP principles and software development best practices. This project serves as an excellent example of building a user-friendly and efficient application while addressing data persistence and error handling. With potential future enhancements, the Console-based Classroom Management System could become a valuable tool for real-world classroom management, benefiting both teachers and students in their academic journeys.

Rubrics

1. User Management (10 points)
    - Proper implementation of user classes and attributes (3 points)
    - Correct use of inheritance and polymorphism for handling user roles (3 points)
    - Effective password management features (encryption, validation) (4 points)
2. Class Management (10 points)
    - Correct representation of class objects and attributes (3 points)

- Proper association between classes and users (aggregation, composition) (3 points)
- Ability to create, update, and delete classes and manage class rosters (4 points)

3. Assignment Management (10 points)
   - Accurate representation of assignment objects and attributes (3 points)
   - Appropriate validation for assignment creation and deadlines (3 points)
   - Ability to manage assignment details and attached files (4 points)
4. Submission and Grading (10 points)
   - Correct implementation of submission objects and attributes (3 points)
   - Proper handling of different submission types (text, file uploads) using OOP concepts (3 points)
   - Validations for submission deadlines and grading (4 points)
5. Gradebook (5 points)
   - Accurate calculation and updating of student grades based on assignment scores and weightage (3 points)
   - Ability to generate grade reports or transcripts (2 points)
6. File Management (5 points)
   - Effective file upload, download, and deletion operations (3 points)
   - Proper enforcement of file size limits and formats (2 points)
7. Notification System (5 points)
   - Implementation of notification generation and delivery (3 points)
   - Customization of notification preferences (2 points)
8. User Interface (5 points)
   - Design of a user-friendly and intuitive console-based UI (3 points)
   - Error handling and informative prompts (2 points)
9. Data Persistence (5 points)
   - Proper file handling techniques for data persistence (5 points)
10. Exception Handling (5 points)
    - Comprehensive and effective handling of runtime errors and exceptional conditions (5 points)
11. Class Diagram (35 points)
    - Accurate representation of classes and their relationships (15 points)
    - Proper use of inheritance and polymorphism in the class diagram (10 points)
    - Appropriate association and aggregation/composition relationships (10 points)

Comments and Code Quality (5 points)

- Proper comments and function naming (3 points)
- Code organization and clarity (2 points)

Bonus Points (5 points)

- Additional features or enhancements beyond the specified requirements (up to 5 points)

Good Luck