Course Project

University Library System

1. Project Description:

The goal is to design and implement an object-oriented program in C++ that simulates the management of a university library. This project allows you to practice designing classes, implementing inheritance, and utilizing various OOP concepts. Functionalities like **create, update, display, and delete** are essential operations for managing objects in an object-oriented program.

2. Project Key Components:

2.1 Classes (MUST implement these classes):

- **Book:** Represents a book with attributes like title, author, ISBN, publication year, genre, availability status (checked out/available), and possibly a due date for checked-out books.
- **Member:** Represents a library member (student, faculty, etc.) with attributes like name, ID, type (student/faculty), list of checked out books, and possible overdue fines.
- **Librarian:** Extends Member with functionalities like adding/removing books, managing member accounts, processing loans/returns, etc.
- **Loan:** Tracks borrowed books with details like member ID, book ID, borrow date, due date, etc.
- **Custom String Class:** Implement this class with appropriate constructors, destructor, display method, concatenation (+) and comparison (==) operators, etc.

2.2 Functionalities:

- **For members:** Search for books (by title or author, etc), view available copies, request loans, return borrowed books, check due dates, manage account details, etc.
- **For librarians:** Add/remove books, update book information, manage member accounts (ability to register new members and remove them), process loan requests (assign books to a member, handle returning a book, updating its availability, etc), generate reports on borrowings and overdue books, etc.

2.3 Implementation:

- Use appropriate data structures (e.g. arrays, vectors, etc.) to store books, members, and loans.
- Implement methods for each class to perform their respective functionalities.

3. General rules

3.1 Programming language used is C++.

3.2 A team can be formed from 5 to 6 students.

4. Suggestions for Students:

- **Start with a basic design:** Begin with essential functionalities like adding/searching books, borrowing/returning books, and managing member information.
- **Incorporate inheritance:** Use inheritance to create different member types (students, faculty, staff) with varying borrowing privileges.
- **Implement data persistence:** Store book and member data in files to persist information even after program termination.
- **Test thoroughly:** Test all functionalities under various scenarios to ensure correct behavior and data integrity.
- **Consider advanced features:** Implement functionalities like book reservations, waitlists, overdue fines, and generating reports on borrowing statistics.
- **Document your code:** Use comments and docstrings to explain code logic and structure, making it easier to understand and maintain.

5. Evaluation Criteria:

5.1 Ideas and Design:

5.1.1 Thoughtful class design and application of OOP concepts **(encapsulation and abstraction are not enough. Using inheritance and polymorphism is mandatory)**.

5.1.2 Clarity of explanations for choices made.

5.2 Implementation:

5.2.1 Correctness of the String class.

5.2.2 Functionality of the library core features (add/remove resources, members, checkout/return, search).

5.2.3 Code quality, readability, and adherence to best practices.

5.3 Documentation:

4.3.1 Well-commented code.

4.3.2 Comprehensive presentation (PowerPoint presentation and poster) explaining the design, key functionalities, choices made, individual contributions of each team member, and a list of all references). **Steps to run the code should be clear (IDE version, any installed packages, etc.)**

4.3.3 Clear walkthrough video demonstrating the working program (3 – 5 mins) and how to run it and how to use it **(Steps to run the code should be clear)**.

6. Deadline and late submission policy

- The deadline for project submission (including all deliverables) is ***Thursday April 25th 2024*** before ***11:59 PM*** **(Cairo time zone)**
- **Late submissions are strictly prohibited.**
- **Project discussion will be held on *Sunday, April 28th 2024*.**

7. Rubrics:

Deliverables (10 points)	**Grading criteria**
C++ code	• All (.cpp and .h) files are delivered and in correct format. • Well-commented code that runs without error. • Excellent use of white space (indentation), naming of variables, methods. • Excellent user prompts and organized output.
Presentation (.ppt)	• Explanation of the design, key functionalities, choices made. • Individual contributions of each team member. • All references used must be included. • University and Faculty logos are included. • Readable presentation using appropriate fonts, size, colors. • Clear and organized presentation.
Poster (.ppt)	• One-page poster. • Clear and concise version of the presentation, keeping only necessary information. • All references used must be included. • University and Faculty logos are included. • Readable poster using appropriate fonts, size, colors.
Walkthrough video	• Clear walkthrough video that showcase the functionality of your project (how to run it and how to use it). • Duration: 3 – 5 mins
Discussion (10 points)	**Grading criteria**
Discussion (oral presentation)	• All team members **must** attend the discussion. • All team members **must** be involved in presenting their work. • Each team member **must** show understanding of the code by giving answers to the asked questions including **modifying the code during discussion.**

Bonus

To be eligible for the **3 bonus** marks, you must incorporate a graphical user interface (GUI) using Java, or Python, or C++ in your project. However, it is important to note that you will only receive the bonus points if you demonstrate a clear understanding of the code.

Note 1:

All deliverables are to be submitted as a single .zip file named as "*Project_TeamLeaderID.zip*", where "*TeamLeaderID*" is the EUI student ID of the leader of the team. Only one member of the team **has** to submit this .zip file on Canvas under the assignment section entitled "Project".

Note 2

All team members **MUST** attend the discussion, as you won't be graded unless you attend it.