RED & WHITE®
Multimedia Education
Shaping "skills" for "scaling" higher...!!!

C++

Project - 5

Library Management System

RED & WHITE MULTIMEDIA EDUCATION

Shaping “skills” for “scaling” higher…!!!

From the Headquarter of RNW
Surat, Gujarat, India
https://www.rnwmultimedia.edu.in

Project Definition: Library Management System

Overview:

Develop a Library Management System (LMS) in C++. The LMS should allow users to manage books, patrons, and transactions. The system should demonstrate the principles of abstraction, encapsulation, inheritance, and polymorphism.

Time Allocation:

- Total Duration: 4 Hours
- Total Marks: 10

Instructions:

1. Attempt all assigned tasks.
2. Make suitable assumptions wherever necessary.
3. Upload your exam task by uploading the project to GitHub and submitting the GitHub repository link which must have screenshots of your output in a README.md file.
4. This project is individual-based; copying code from classmates is prohibited.

Remember to follow the instructions provided professionally, make suitable assumptions wherever necessary, and avoid copying code or content from any unauthorized sources. Good luck with your project work!

Project Criteria:

Requirements:

- Class & Object
- Abstraction
- Polymorphism
- Inheritance
- Encapsulation

Abstract Base Class (LibraryItem):

- The LibraryItem class serves as an abstract base class for all library items (e.g., books, DVDs, magazines).
- It defines common attributes (e.g., title, author, due date) and operations (e.g., checking out, returning).
- The following pure virtual functions should be declared within LibraryItem:
  - virtual void checkOut() = 0;
  - virtual void returnItem() = 0;
  - virtual void displayDetails() = 0;

Exception Handling:

- Implement error handling for scenarios such as invalid input (e.g., negative book quantities, incorrect ISBN format).
- Use general exception to handle other exceptions (if any).

Encapsulation:

Each LibraryItem should encapsulate its data members:

- **Private data members (attributes):**
    - string title
    - string author
    - string dueDate
- **Public member functions (getters and setters):**
    - getTitle()
    - getAuthor()
    - getDueDate()
    - setTitle(string newTitle)
    - setAuthor(string newAuthor)
    - setDueDate(string newDueDate)

Inheritance:

Derived Classes (e.g., Book, DVD, Magazine):

- Each derived class should inherit from LibraryItem.
- Additional attributes specific to each type (e.g., duration for DVDs, issueNumber for magazines) can be added.
- Implement the pure virtual functions in each derived class:
    - void checkOut() override;
    - void returnItem() override;
    - void displayDetails() const override;

Polymorphism:

Demonstrate polymorphism by creating an array of LibraryItem pointers:

- LibraryItem* libraryItems[MAX_ITEMS];
- Populate the array with instances of different item types (e.g., books, DVDs).
- Invoke common methods (e.g., checkOut(), returnItem()) on different item types.

Pointer Object:

Use dynamic memory allocation (pointers) to manage library items:

- When adding items to the catalog, allocate memory dynamically (e.g., new Book, new DVD).

- Properly release memory using delete when items are removed.

User Interaction:

- Create a menu-driven console interface for users to interact with the LMS.
- Allow users to search for items, check out, and return them.

Marking Criteria: (Total 10 Marks)

Logic: 8 Mark

Output: 2 Mark