Library Management System

Objective:
Design and implement a Library Management System that efficiently manages books, members, and their interactions using object-oriented principles, including classes and inheritance.
Problem Description:
The Library Management System will manage the books in the library, handle membership, and allow users to borrow and return books. The system will be designed using object-oriented principles, particularly focusing on the use of classes and inheritance to represent different entities in the library.

Key Components:
1. Base Class: `LibraryItem`
   - Attributes:
     - `title`: The title of the item.
     - `uniqueID`: A unique identifier for the item.
     - `isAvailable`: A boolean flag to indicate whether the item is available for borrowing.
   - Methods:
     - `checkOut()`: Marks the item as checked out.
     - `returnItem()`: Marks the item as available.
     - `displayInfo()`: Displays the item’s details.
   - Description: The `LibraryItem` class will be a base class representing any item in the library. This class will provide common attributes and methods that can be inherited by more specific classes.

2. Derived Classes:
   - `Book` Class (inherits from `LibraryItem`):
     - Additional Attributes:
       - `author`: The author of the book.
       - `genre`: The genre of the book.
       - `pages`: The number of pages in the book.
     - Additional Methods:
       - `displayBookInfo()`: Displays detailed information about the book, including author and genre.
     - Description: The `Book` class extends `LibraryItem` and includes additional attributes and methods specific to books.
   - `Magazine` Class (inherits from `LibraryItem`):
     - Additional Attributes:
       - `issueNumber`: The issue number of the magazine.
       - `publicationDate`: The publication date of the magazine.
     - Additional Methods:
       - `displayMagazineInfo()`: Displays details specific to the magazine, like issue number and publication date.
     - Description: The `Magazine` class extends `LibraryItem` and includes attributes and methods specific to magazines.

3. Member Class:
  - Attributes:
    - `name`: The name of the library member.
    - `memberID`: A unique identifier for the member.
    - `borrowedItems`: A list of items the member has currently borrowed.
  - Methods:
    - `borrowItem(LibraryItem item)`: Allows the member to borrow an item.
    - `returnItem(LibraryItem item)`: Allows the member to return a borrowed item.
    - `displayMemberInfo()`: Displays the member's details and borrowed items.
  - Description: The `Member` class represents a library member who can borrow items. It interacts with instances of the `LibraryItem` and its derived classes.

4. Library Class:
  - Attributes:
    - `libraryItems`: A collection (e.g., ArrayList) of all library items (books, magazines, etc.).
    - `members`: A collection of all registered members.
  - Methods:
    - `addItem(LibraryItem item)`: Adds a new item to the library's inventory.
    - `registerMember(Member member)`: Registers a new member.
    - `findItem(String title)`: Searches for an item by title and returns the item if found.
    - `listAvailableItems()`: Lists all items that are currently available for borrowing.
    - `displayAllMembers()`: Displays a list of all registered members.
  - Description: The `Library` class manages the collection of library items and registered members. It facilitates the borrowing and returning of items and provides functionalities to add new items and members to the system.

5. Library Management System Launcher:
  - Purpose: Implement a `LibraryLauncher` class with a `main()` method to simulate the library system. This class will:
    - Add various library items (books and magazines) to the library.
    - Register members and allow them to borrow and return items.
    - Display the inventory and member details at different stages of the simulation.

Example Usage:

- Create instances of `Book` and `Magazine` and add them to the library.
- Register members using the `Member` class.
- Allow members to borrow and return items using methods in the `Library` and `Member` classes.
- Use inheritance to manage different types of library items while reusing common functionality defined in the `LibraryItem` base class.

This problem statement encourages the use of inheritance to manage different types of library items, while also implementing a system that can be expanded with more item types in the future, demonstrating the power and flexibility of object-oriented programming.