

Q. Design and implement a Library Management System using object-oriented programming principles in Python.

Requirements:

1. Book Class:

- Attributes:
 - title (string)
 - author (string)
 - isbn (string)
 - is_available (boolean, defaults to True)
- Methods:
 - `__init__(self, title, author, isbn)`
 - `__str__(self)`
 - `check_out(self)` - Sets is_available to False
 - `return_book(self)` - Sets is_available to True

2. Member Class:

- Attributes:
 - name (string)
 - member_id (int)
 - borrowed_books (list of Book objects)
- Methods:
 - `__init__(self, name, member_id)`
 - `__str__(self)`
 - `borrow_book(self, book)` - Adds a book to borrowed_books if it is available
 - `return_book(self, book)` - Removes a book from borrowed_books

3. Library Class:

- Attributes:
 - name (string)
 - books (list of Book objects)
 - members (list of Member objects)
- Methods:
 - `__init__(self, name)`
 - `add_book(self, book)` - Adds a book to the library
 - `remove_book(self, book)` - Removes a book from the library
 - `register_member(self, member)` - Registers a new member
 - `find_book_by_title(self, title)` - Returns a list of books with matching title
 - `find_book_by_author(self, author)` - Returns a list of books with matching author
 - `display_available_books(self)` - Returns a list of available books

4. Librarian Class (inherits from Member):

- Methods:
 - `add_book_to_library(self, library, book)` - Adds a book to the library
 - `remove_book_from_library(self, library, book)` - Removes a book from the library

5. Admin Class (inherits from Librarian):

- Methods:

- `view_all_members(self, library)` - Returns a list of all members
- `view_all_books(self, library)` - Returns a list of all books

Task:

1. Implement the `Book`, `Member`, `Library`, `Librarian`, and `Admin` classes with the specified attributes and methods.
2. Create a small script to demonstrate the functionality of the Library Management System.

Example Usage:

- Create a library, add books to the library, register members, and demonstrate borrowing and returning books.
- Use the `Librarian` and `Admin` classes to add or remove books and view lists of members and books.

Submission:

Submit your Python script implementing the above classes and demonstrating their functionality through a series of interactions.

Note: Ensure proper encapsulation and use of OOP principles throughout your implementation.