Assignment: Console-Based Library Management System (LMS)

Overview:

Your task is to design and implement a console-based Library Management System for managing book loans, returns, and user registrations in a library. The system should efficiently manage different types of books and users, and track transactions between them. The program must be written in Java and should showcase a comprehensive understanding of Object-Oriented Programming (OOP) principles such as encapsulation, abstraction, polymorphism, and inheritance, in addition to implementing interfaces and using composition or aggregation.

System Requirements:

User Types:

- **Properties**: User ID, Name, Email, List of Loaned Books, Total Loan Fees, Phone Number, Address
- **Student**:
  - **Features**: Can borrow up to 5 books.
- **Faculty**:
  - **Features**: Can borrow up to 10 books, includes research staff.
- **Public Member**:
  - **Features**: Can borrow up to 3 books, higher base fee.

Book Types:

- **Properties**: Book ID, Title, Author, ISBN, Publication Year, Genre, Loan Status, Base Loan Fee
- **Textbook**:
  - **Features**: Used for academic purposes, often required for a semester.
  - **Loan Calculation Formula**: Base fee + duration-based cost.
  - **Extendable**: Yes (Maximum one extension allowed).
- **Novel**:
  - **Features**: General reading, various genres available.
  - **Loan Calculation Formula**: Flat rate.
  - **Extendable**: No.
- **Reference Book**:
  - **Features**: Used for in-library reading only.
  - **Loan Calculation Formula**: Not applicable (cannot be loaned out).
  - **Extendable**: Not applicable.

Functionalities of LMS:

- **Book Management**:
  - Add new books of different types.
  - Display available books.
  - Remove a book if it is not loaned.
- **User Management**:
  - Add new users of different types.

    - Display user details.
- **Loan Transactions**:
    - Loan a book to a user.
    - Display loan details.
    - Calculate and display the total loan cost.
    - Manage loan extensions and calculate additional fees if applicable.

Damage and Late Return Penalties:

- **For All Books**: Late return fee is a flat rate per day, decided by the library.
- **For Textbooks**: Additional extension fee if extended beyond the initial loan period.

Constraints:

- Implement **encapsulation** to protect the internal state of classes.
- Use **abstraction** to hide complex implementations.
- Utilize **polymorphism** to handle different loan calculations.
- Implement **inheritance** for creating a base class for books and users.
- Implement an **interface** for loanable items that defines methods related to loan operations.
- Demonstrate **composition** or **aggregation** in managing relationships between books and users.
- Identify and use **static** and **final** variables appropriately.
- Use Java Collections Framework for managing groups of objects (e.g., ArrayList, HashMap).