## Project 1 Team: Arshia, Mahan, Saba, Paria, AmirMahdi

**1. Arshia: Define Book Class**

- **Tasks:**
    - Implement the Book class with attributes such as title, author, genre, and quantity.
    - Include methods to manage book quantities and retrieve book information.
- **Responsibilities:**
    - Define and maintain the Book class.
    - Implement methods to increase/decrease quantity and represent the book as a string.

**2. Mahan: Define Library Class and Its Core Methods**

- **Tasks:**
    - Implement the Library class with methods to add books, search books, and manage rentals.
    - Include functionality to add books, increase book count, and search for books.
- **Responsibilities:**
    - Define and maintain the Library class.
    - Implement methods for managing books and rentals within the library.

**3. Saba: Define Member Class**

- **Tasks:**
    - Implement the Member class with attributes and methods related to member activities like renting and returning books.
- **Responsibilities:**
    - Define and maintain the Member class.
    - Implement methods for managing member interactions with the library system, including renting and returning books.

**4. AmirMahdi: Define Rental Class and Rental Management**

- **Tasks:**
    - Implement the Rental class to manage book rentals, track due dates, and calculate late fees.
    - Manage the relationship between Book and Member through the Rental class.
- **Responsibilities:**

- Define and maintain the Rental class.
- Implement methods for tracking rental status, calculating late fees, and updating rental records.

**5. Paria: Define Menu System, Integration, and Testing**

- **Tasks:**
  - Implement a menu system that allows library staff and members to perform their respective actions.
  - Handle user input to simulate interactions with the system.
  - Integrate all components and conduct testing to ensure the system works as expected.
- **Responsibilities:**
  - Implement the menu system for library staff and member actions.
  - Handle user input and direct actions based on the menu options.
  - Conduct testing and ensure the integrated functionality works correctly.

**Class Breakdown and Relationships:**

**1. Book**

- **Attributes:** title, author, genre, quantity
- **Methods:** get_title(), get_author(), get_genre(), get_quantity(), increase_quantity(amount), decrease_quantity(), __str__()
- **Relationship:**
  - Used by Member to manage rented books.
  - Managed by Library for book inventory.

**2. Member**

- **Attributes:** member_id, name, rented_books
- **Methods:** get_member_id(), get_name(), get_rented_books(), rent_book(book), return_book(book), __str__()
- **Relationship:**
  - Interacts with Book through renting and returning books.
  - Managed by Library for member-related operations.

**3. Rental**

- **Attributes:** book, member, rented_on, due_date, returned_on

- **Methods:** get_due_date(), get_returned_on(), set_returned_on(returned_on), is_late(), calculate_fee(), __str__()
- **Relationship:**
    - Connects Book and Member for tracking the rental.
    - Managed by Library for tracking rental status and fees.

**4. Library**

- **Attributes:** books, members, rented_books
- **Methods:** add_book(book), increase_book_count(book_title, quantity), search_book(title, author, genre), add_member(member), rent_book(book_title, member_id), get_rental_status(), evaluate_late_fee(days_late)
- **Relationship:**
    - Manages collections of Book and Member.
    - Tracks Rental objects to manage book rentals and return status.

**5. Menu System**

- **Responsibilities:**
    - Manages the interface for interactions between Library, Member, and Rental classes.
    - Provides a command-line interface for users to interact with the library system.
    - Ensures proper input handling, action delegation, and system testing.