

```
In [1]: class Book:
    total_books = 0

    def __init__(self, title, author, isbn):
        self.title = title
        self.author = author
        self.isbn = isbn
        Book.total_books += 1

    def update_title(self, new_title):
        self.title = new_title

    def update_author(self, new_author):
        self.author = new_author

    def display_info(self, user_type="reader"):
        if user_type == "librarian":
            return f"[Librarian View] Title: {self.title}, Author: {self.author}"
        else:
            return f"Title: {self.title}, Author: {self.author}"

    @staticmethod
    def book_info():
        return "Books contain titles, authors, and ISBNs."

    @classmethod
    def get_total_books(cls):
        return cls.total_books

class Author:
    total_authors = 0

    def __init__(self, name, birthdate):
        self.name = name
        self.birthdate = birthdate
        self.books = []
        Author.total_authors += 1

    def add_book(self, book):
        if isinstance(book, Book):
            self.books.append(book)

    def remove_book(self, isbn):
        self.books = [book for book in self.books if book.isbn != isbn]

    @staticmethod
    def author_info():
        return "Authors write books and have a name and birthdate."

    @classmethod
    def get_total_authors(cls):
        return cls.total_authors

class Library:
    library_count = 0
```

```

def __init__(self):
    self.books = []
    self.authors = []
    Library.library_count += 1

def add_book(self, book):
    if isinstance(book, Book):
        self.books.append(book)

def remove_book(self, isbn):
    self.books = [book for book in self.books if book.isbn != isbn]

def list_books(self):
    for book in self.books:
        print(book.display_info())

def add_author(self, author):
    if isinstance(author, Author):
        self.authors.append(author)

    @staticmethod
    def library_info():
        return "A library contains books and authors."

    @classmethod
    def get_library_count(cls):
        return cls.library_count

author1 = Author("J.K. Rowling", "1965-07-31")
author2 = Author("George Orwell", "1903-06-25")

book1 = Book("Harry Potter", "J.K. Rowling", "1234567890")
book2 = Book("1984", "George Orwell", "9876543210")

author1.add_book(book1)
author2.add_book(book2)

library = Library()

library.add_book(book1)
library.add_book(book2)
library.add_author(author1)
library.add_author(author2)

print("\nBooks in Library:")
library.list_books()

library.remove_book("9876543210")

print("\nBooks in Library after removing 1984:")
library.list_books()

print("\nBook info for reader:", book1.display_info("reader"))
print("Book info for librarian:", book1.display_info("librarian"))

print("\nStatic Info:")
print(Book.book_info())
print(Author.author_info())
print(Library.library_info())

```

```
print("\nClass Counts:")
print("Total Books:", Book.get_total_books())
print("Total Authors:", Author.get_total_authors())
print("Total Libraries:", Library.get_library_count())
```

Books in Library:

Title: Harry Potter, Author: J.K. Rowling

Title: 1984, Author: George Orwell

Books in Library after removing 1984:

Title: Harry Potter, Author: J.K. Rowling

Book info for reader: Title: Harry Potter, Author: J.K. Rowling

Book info for librarian: [Librarian View] Title: Harry Potter, Author: J.K. Rowling, ISBN: 1234567890

Static Info:

Books contain titles, authors, and ISBNs.

Authors write books and have a name and birthdate.

A library contains books and authors.

Class Counts:

Total Books: 2

Total Authors: 2

Total Libraries: 1

In []: