```python
In [1]: #Library Management System
        class Book:
            total_books = 0  # class variable

            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

            # Simulated Method Overloading
            def display_info(self, user_type="reader"):
                if user_type.lower() == "librarian":
                    print(f"Title: {self.title}, Author: {self.author}, ISBN: {self.isbn
                elif user_type.lower() == "reader":
                    print(f"Title: {self.title}, Author: {self.author}")
                else:
                    print("Invalid user type. Showing basic info.")
                    print(f"Title: {self.title}")

            @staticmethod
            def book_info():
                print("Books contain knowledge and are written by authors. Each book has

            @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 = []  # list of Book instances
                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():
                print("Authors write books and contribute to literature and knowledge.")

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

```python
class Library:
    library_count = 0

    def __init__(self):
        self.books = []     # list of Book instances
        self.authors = []   # list of Author instances
        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):
        if not self.books:
            print("No books in the library.")
        for book in self.books:
            book.display_info("reader")

    @staticmethod
    def library_info():
        print("Libraries store books and provide access to readers and researche

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


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

# Create Books
book1 = Book("Harry Potter and the Philosopher's Stone", "J.K. Rowling", "978074
book2 = Book("1984", "George Orwell", "9780451524935")
book3 = Book("Animal Farm", "George Orwell", "9780451526342")

# Add Books to Authors
author1.add_book(book1)
author2.add_book(book2)
author2.add_book(book3)

# Create Library
library1 = Library()

# Add Books to Library
library1.add_book(book1)
library1.add_book(book2)
library1.add_book(book3)

# Display Books in Library
print("\n--- Library Book List ---")
library1.list_books()

# Update a Book Title
book3.update_title("Animal Farm: A Fairy Story")
```

```python
# Simulate Method Overloading
print("\n--- Book Info for Librarian ---")
book2.display_info("librarian")

print("\n--- Book Info for Reader ---")
book2.display_info("reader")

# Remove Book by ISBN from Library and Author
library1.remove_book("9780451524935")
author2.remove_book("9780451524935")

# List books again
print("\n--- Library Book List After Removal ---")
library1.list_books()

# Static Method Calls
print("\n--- Static Method Info ---")
Book.book_info()
Author.author_info()
Library.library_info()

# Class Method Calls
print("\n--- Class Method Counts ---")
print("Total Books:", Book.get_total_books())
print("Total Authors:", Author.get_total_authors())
print("Total Libraries:", Library.get_library_count())
```

```
--- Library Book List ---
Title: Harry Potter and the Philosopher's Stone, Author: J.K. Rowling
Title: 1984, Author: George Orwell
Title: Animal Farm, Author: George Orwell

--- Book Info for Librarian ---
Title: 1984, Author: George Orwell, ISBN: 9780451524935

--- Book Info for Reader ---
Title: 1984, Author: George Orwell

--- Library Book List After Removal ---
Title: Harry Potter and the Philosopher's Stone, Author: J.K. Rowling
Title: Animal Farm: A Fairy Story, Author: George Orwell

--- Static Method Info ---
Books contain knowledge and are written by authors. Each book has a unique ISBN.
Authors write books and contribute to literature and knowledge.
Libraries store books and provide access to readers and researchers.

--- Class Method Counts ---
Total Books: 3
Total Authors: 2
Total Libraries: 1
```