CODE:

public class Library {

privateArrayList<Book>allBook = new ArrayList<Book>();

public Library(ArrayList<Book> other) {

if (other == null) {

throw new NullPointerException("null pointer");

} else

this.allBook = other;

}

public Library() {

this.allBook = new ArrayList<Book>();

}

publicboolean add(Book book) {

if (book != null && !book.equals("")) {

throw new IllegalArgumentException("Can't be empty");

}

allBook.add(book);

return true;

}

publicArrayList<Book>findTitles(String title) {

for(Book b: allBook) {

if(title.compareTo(b.getTitle())== 0) {

returnallBook;

}

}

return null;

}

public void sort() {

Collections.sort(allBook);

}

public String toString() {

returnLibrary.this.toString();

}

}

public class Book implements Comparable<Book> {

private String bookTitle;

privateArrayList<String>bookAuthor;

public Book(String title, ArrayList<String> authors) {

if(title == null && authors == null) {

throw new IllegalArgumentException("Can't be null");

}

if(title.isEmpty() &&authors.isEmpty()) {

throw new IllegalArgumentException("Can't be empty");

}

bookTitle = title;

bookAuthor = authors;

}

public String getTitle() {

returnbookTitle;

}

publicArrayList<String>getAuthors( ) {

returnbookAuthor;

}

public String toString( ) {

returnbookTitle + bookAuthor;

}

publicintcompareTo(Book other){

returnbookTitle.compareTo(other.bookTitle);

}

publicboolean equals(Object o) {

if(!(o instanceof Book)) {

return false;

}

Book b = (Book) o;

returnb.bookTitle.equals(bookTitle)

&&b.bookAuthor.equals(bookAuthor);

}

}