



# K.R. Mangalam University

## School of Engineering & Technology

Fundamentals Of Java Programming

(ENCA203) Assignment 4

City Library Digital Management System.

Submitted by:

Name: RAKESH G  
Kumar

Roll No: 2401201064

Class: BCA (AI & DS)

Submitted to:

Dr. Manish

GitHub Repository:

[https://github.com/rakesh4407/Java\\_Assignment\\_Rakesh](https://github.com/rakesh4407/Java_Assignment_Rakesh)

# Code:

The screenshot shows two tabs open in a Java code editor: `Rakesh_Assingment.ipynb` and `ResultManager.java 7`. The `LibrarySystem.java 9+` tab is the active one, showing the following code:

```
1 import java.io.*;
2 import java.util.*;
3
4 public class LibrarySystem {
5
6     static class Book implements Comparable<Book> {
7
8         private int bookId; Field bookId can be final
9         private String title; Field title can be final
10        private String author; Field author can be final
11        private String category; Field category can be final
12        private boolean isIssued;
13
14        public Book(int bookId, String title, String author, String category) {
15            this.bookId = bookId;
16            this.title = title;
17            this.author = author;
18            this.category = category;
19            this.isIssued = false;
20        }
21
22        public int getBookId() {
23            return bookId;
24        }
25
26        public String getTitle() {
27            return title;
28        }
29
30        public String getAuthor() {
31            return author;
32        }
33
34        public String getCategory() {
35            return category;
36        }
37
38        public boolean isIssued() {
39            return isIssued;
40        }
41
42        public void markAsIssued() {
43            isIssued = true;
44        }
45    }
46}
```

The screenshot shows two tabs open in a Java code editor: `Rakesh_Assingment.ipynb` and `ResultManager.java 7`. The `LibrarySystem.java 9+` tab is the active one, showing the following code:

```
5 public class LibrarySystem {
7     static class Book implements Comparable<Book> {
47         public void markAsReturned() {
48             isIssued = false;
49         }
50
51         public void displayBookDetails() {
52             System.out.println("Book ID: " + bookId);
53             System.out.println("Title: " + title);
54             System.out.println("Author: " + author);
55             System.out.println("Category: " + category);
56             System.out.println("Issued: " + (isIssued ? "Yes" : "No"));
57         }
58
59         @Override
60         public int compareTo(Book other) {
61             return this.title.compareToIgnoreCase(other.title);
62         }
63
64         @Override
65         public String toString() {
66             return bookId + "," + title + "," + author + "," + category + "," + isIssued;
67         }
68
69         public static Book fromString(String line) {
70             String[] p = line.split(",");
71             if (p.length == 5) {
72                 Book b = new Book(Integer.parseInt(p[0]), p[1], p[2], p[3]);
73                 b.isIssued = Boolean.parseBoolean(p[4]);
74                 return b;
75             }
76             return null;
77         }
78     }
79
80     static class Member {
81
82         private int memberId; Field memberId can be final
83         private String name; Field name can be final
84         private String email; Field email can be final
85         private List<Integer> issuedBooks = new ArrayList<>(); Field issuedBooks can be final
86
87         public Member(int memberId, String name, String email) {
88             this.memberId = memberId;
89             this.name = name;
90             this.email = email;
91         }
92     }
93 }
```

```

Rakesh_Assingment.ipynb  ResultManager.java 7  LibrarySystem.java 9+ X
JAVA > LibrarySystem.java > Language Support for Java(TM) by Red Hat > LibrarySystem > Book > getBookId()
5  public class LibrarySystem {
80   static class Member {
87     public Member(int memberId, String name, String email) {
91   }
92
93     public int getMemberId() {
94       return memberId;
95     }
96
97     public String getName() {
98       return name;
99     }
100
101    public String getEmail() {
102      return email;
103    }
104
105    public void addIssuedBook(int bookId) {
106      issuedBooks.add(bookId);
107    }
108
109    public void returnIssuedBook(int bookId) {
110      issuedBooks.remove(Integer.valueOf(bookId));
111    }
112
113    public void displayMemberDetails() {
114      System.out.println("Member ID: " + memberId);
115      System.out.println("Name: " + name);
116      System.out.println("Email: " + email);
117      System.out.println("Issued Books: " + issuedBooks);
118    }
119
120  @Override
121  public String toString() {
122    StringBuilder sb = new StringBuilder();
123    sb.append(memberId).append(".").append(name).append(",").append(email).append(",");
124    for (int i = 0; i < issuedBooks.size(); i++) {
125      sb.append(issuedBooks.get(i));
126      if (i < issuedBooks.size() - 1) {
127        sb.append(";");
128      }
129    }
130    return sb.toString();
131  }
132
133  public static Member fromString(String line) {
Rakesh_Assingment.ipynb  ResultManager.java 7  LibrarySystem.java 9+ X
JAVA > LibrarySystem.java > Language Support for Java(TM) by Red Hat > LibrarySystem > Book > getBookId()
5  public class LibrarySystem {
80   static class Member {
83     public static Member fromString(String line) {
134       String[] p = line.split(',', 4);
135       if (p.length >= 3) {
136         Member m = new Member(Integer.parseInt(p[0]), p[1], p[2]);
137         if (p.length == 4 && p[3].isEmpty()) {
138           String[] books = p[3].split(";");
139           for (String b : books) {
140             m.issuedBooks.add(Integer.parseInt(b));  Unnecessary temporary when converting from String
141           }
142         }
143       }
144       return m;
145     }
146     return null;
147   }
148
149   static class LibraryManager {
150
151     private Map<Integer, Book> books = new HashMap<>();  Field books can be final
152     private Map<Integer, Member> members = new HashMap<>();  Field members can be final
153     private final String BOOK_FILE = "books.txt";
154     private final String MEMBER_FILE = "members.txt";
155     private Scanner sc = new Scanner(System.in);  Field sc can be final
156
157     public LibraryManager() {
158       loadFromFile();
159     }
160
161     private void saveBooks() {
162       try (BufferedWriter bw = new BufferedWriter(new FileWriter(BOOK_FILE))) {
163         for (Book b : books.values()) {
164           bw.write(b.toString());
165           bw.newLine();
166         }
167       } catch (IOException e) {
168         System.out.println("Error saving books: " + e.getMessage());
169       }
170     }
171
172     private void saveMembers() {
173       try (BufferedWriter bw = new BufferedWriter(new FileWriter(MEMBER_FILE))) {
174         for (Member m : members.values()) {
175           bw.write(m.toString());
176           bw.newLine();
177         }
178       }
179     }
180   }
181
182   public void loadFromFile() {
183     books.clear();
184     members.clear();
185     try (Scanner s = new Scanner(new File(BOOK_FILE))) {
186       while (s.hasNextLine()) {
187         String line = s.nextLine();
188         String[] parts = line.split(",");
189         if (parts.length == 4) {
190           int id = Integer.parseInt(parts[0]);
191           String title = parts[1];
192           String author = parts[2];
193           String status = parts[3];
194           Book book = new Book(id, title, author);
195           if ("ISSUED".equals(status)) {
196             book.setStatus(true);
197           } else {
198             book.setStatus(false);
199           }
200           books.put(id, book);
201         }
202       }
203     } catch (FileNotFoundException e) {
204       System.out.println("Error loading books file: " + e.getMessage());
205     }
206   }
207
208   public void saveToFile() {
209     try (BufferedWriter bw = new BufferedWriter(new FileWriter(BOOK_FILE))) {
210       for (Book b : books.values()) {
211         bw.write(b.toString());
212         bw.newLine();
213       }
214     } catch (IOException e) {
215       System.out.println("Error saving books file: " + e.getMessage());
216     }
217   }
218
219   public void saveMembersToFile() {
220     try (BufferedWriter bw = new BufferedWriter(new FileWriter(MEMBER_FILE))) {
221       for (Member m : members.values()) {
222         bw.write(m.toString());
223         bw.newLine();
224       }
225     } catch (IOException e) {
226       System.out.println("Error saving members file: " + e.getMessage());
227     }
228   }
229
230   public void displayBooks() {
231     System.out.println("Books in the library:");
232     for (Book b : books.values()) {
233       System.out.println(b);
234     }
235   }
236
237   public void displayMembers() {
238     System.out.println("Members in the library:");
239     for (Member m : members.values()) {
240       System.out.println(m);
241     }
242   }
243
244   public void searchBook(String title) {
245     Book foundBook = null;
246     for (Book b : books.values()) {
247       if (b.getTitle().equalsIgnoreCase(title)) {
248         foundBook = b;
249         break;
250       }
251     }
252     if (foundBook != null) {
253       System.out.println("Book found: " + foundBook);
254     } else {
255       System.out.println("Book not found: " + title);
256     }
257   }
258
259   public void searchMember(String name) {
260     Member foundMember = null;
261     for (Member m : members.values()) {
262       if (m.getName().equalsIgnoreCase(name)) {
263         foundMember = m;
264         break;
265       }
266     }
267     if (foundMember != null) {
268       System.out.println("Member found: " + foundMember);
269     } else {
270       System.out.println("Member not found: " + name);
271     }
272   }
273
274   public void issueBook(int memberId, int bookId) {
275     Member member = members.get(memberId);
276     Book book = books.get(bookId);
277     if (member != null && book != null && !book.isIssued()) {
278       member.addIssuedBook(bookId);
279       book.setStatus(true);
280       System.out.println("Book issued successfully.");
281     } else {
282       System.out.println("Member or book not found.");
283     }
284   }
285
286   public void returnBook(int memberId, int bookId) {
287     Member member = members.get(memberId);
288     Book book = books.get(bookId);
289     if (member != null && book != null && book.isIssued()) {
290       member.returnIssuedBook(bookId);
291       book.setStatus(false);
292       System.out.println("Book returned successfully.");
293     } else {
294       System.out.println("Member or book not found.");
295     }
296   }
297
298   public void displayIssuedBooks() {
299     System.out.println("Issued Books:");
300     for (Member m : members.values()) {
301       if (m.isIssued()) {
302         System.out.println(m);
303       }
304     }
305   }
306
307   public void displayAvailableBooks() {
308     System.out.println("Available Books:");
309     for (Book b : books.values()) {
310       if (!b.isIssued()) {
311         System.out.println(b);
312       }
313     }
314   }
315
316   public void displayMemberDetails(int memberId) {
317     Member member = members.get(memberId);
318     if (member != null) {
319       System.out.println(member);
320     } else {
321       System.out.println("Member not found: " + memberId);
322     }
323   }
324
325   public void displayMemberBooks(int memberId) {
326     Member member = members.get(memberId);
327     if (member != null) {
328       System.out.println("Books issued by Member ID " + memberId + ":");
329       for (Book b : member.getIssuedBooks()) {
330         System.out.println(b);
331       }
332     } else {
333       System.out.println("Member not found: " + memberId);
334     }
335   }
336
337   public void displayMemberEmails(int memberId) {
338     Member member = members.get(memberId);
339     if (member != null) {
340       System.out.println("Emails of Member ID " + memberId + ":");
341       for (String email : member.getEmails()) {
342         System.out.println(email);
343       }
344     } else {
345       System.out.println("Member not found: " + memberId);
346     }
347   }
348
349   public void displayMemberNames(int memberId) {
350     Member member = members.get(memberId);
351     if (member != null) {
352       System.out.println("Names of Member ID " + memberId + ":");
353       for (String name : member.getNames()) {
354         System.out.println(name);
355       }
356     } else {
357       System.out.println("Member not found: " + memberId);
358     }
359   }
360
361   public void displayMemberIds(int memberId) {
362     Member member = members.get(memberId);
363     if (member != null) {
364       System.out.println("IDs of Member ID " + memberId + ":");
365       for (int id : member.getIds()) {
366         System.out.println(id);
367       }
368     } else {
369       System.out.println("Member not found: " + memberId);
370     }
371   }
372
373   public void displayMemberStatus(int memberId) {
374     Member member = members.get(memberId);
375     if (member != null) {
376       System.out.println("Status of Member ID " + memberId + ":");
377       for (String status : member.getStatuses()) {
378         System.out.println(status);
379       }
380     } else {
381       System.out.println("Member not found: " + memberId);
382     }
383   }
384
385   public void displayMemberTitles(int memberId) {
386     Member member = members.get(memberId);
387     if (member != null) {
388       System.out.println("Titles of Member ID " + memberId + ":");
389       for (String title : member.getTitles()) {
390         System.out.println(title);
391       }
392     } else {
393       System.out.println("Member not found: " + memberId);
394     }
395   }
396
397   public void displayMemberAuthors(int memberId) {
398     Member member = members.get(memberId);
399     if (member != null) {
400       System.out.println("Authors of Member ID " + memberId + ":");
401       for (String author : member.getAuthors()) {
402         System.out.println(author);
403       }
404     } else {
405       System.out.println("Member not found: " + memberId);
406     }
407   }
408
409   public void displayMemberStatuses(int memberId) {
410     Member member = members.get(memberId);
411     if (member != null) {
412       System.out.println("Statuses of Member ID " + memberId + ":");
413       for (String status : member.getStatuses()) {
414         System.out.println(status);
415       }
416     } else {
417       System.out.println("Member not found: " + memberId);
418     }
419   }
420
421   public void displayMemberTitlesAndAuthors(int memberId) {
422     Member member = members.get(memberId);
423     if (member != null) {
424       System.out.println("Titles and Authors of Member ID " + memberId + ":");
425       for (String title : member.getTitles()) {
426         System.out.println(title);
427       }
428       for (String author : member.getAuthors()) {
429         System.out.println(author);
430       }
431     } else {
432       System.out.println("Member not found: " + memberId);
433     }
434   }
435
436   public void displayMemberNamesAndEmails(int memberId) {
437     Member member = members.get(memberId);
438     if (member != null) {
439       System.out.println("Names and Emails of Member ID " + memberId + ":");
440       for (String name : member.getNames()) {
441         System.out.println(name);
442       }
443       for (String email : member.getEmails()) {
444         System.out.println(email);
445       }
446     } else {
447       System.out.println("Member not found: " + memberId);
448     }
449   }
450
451   public void displayMemberIdsAndTitles(int memberId) {
452     Member member = members.get(memberId);
453     if (member != null) {
454       System.out.println("IDs and Titles of Member ID " + memberId + ":");
455       for (int id : member.getIds()) {
456         System.out.println(id);
457       }
458       for (String title : member.getTitles()) {
459         System.out.println(title);
460       }
461     } else {
462       System.out.println("Member not found: " + memberId);
463     }
464   }
465
466   public void displayMemberNamesAndAuthors(int memberId) {
467     Member member = members.get(memberId);
468     if (member != null) {
469       System.out.println("Names and Authors of Member ID " + memberId + ":");
470       for (String name : member.getNames()) {
471         System.out.println(name);
472       }
473       for (String author : member.getAuthors()) {
474         System.out.println(author);
475       }
476     } else {
477       System.out.println("Member not found: " + memberId);
478     }
479   }
480
481   public void displayMemberEmailsAndTitles(int memberId) {
482     Member member = members.get(memberId);
483     if (member != null) {
484       System.out.println("Emails and Titles of Member ID " + memberId + ":");
485       for (String email : member.getEmails()) {
486         System.out.println(email);
487       }
488       for (String title : member.getTitles()) {
489         System.out.println(title);
490       }
491     } else {
492       System.out.println("Member not found: " + memberId);
493     }
494   }
495
496   public void displayMemberNamesAndEmailsAndTitles(int memberId) {
497     Member member = members.get(memberId);
498     if (member != null) {
499       System.out.println("Names, Emails, and Titles of Member ID " + memberId + ":");
500       for (String name : member.getNames()) {
501         System.out.println(name);
502       }
503       for (String email : member.getEmails()) {
504         System.out.println(email);
505       }
506       for (String title : member.getTitles()) {
507         System.out.println(title);
508       }
509     } else {
510       System.out.println("Member not found: " + memberId);
511     }
512   }
513
514   public void displayMemberNamesAndAuthorsAndTitles(int memberId) {
515     Member member = members.get(memberId);
516     if (member != null) {
517       System.out.println("Names, Authors, and Titles of Member ID " + memberId + ":");
518       for (String name : member.getNames()) {
519         System.out.println(name);
520       }
521       for (String author : member.getAuthors()) {
522         System.out.println(author);
523       }
524       for (String title : member.getTitles()) {
525         System.out.println(title);
526       }
527     } else {
528       System.out.println("Member not found: " + memberId);
529     }
530   }
531
532   public void displayMemberNamesAndEmailsAndAuthorsAndTitles(int memberId) {
533     Member member = members.get(memberId);
534     if (member != null) {
535       System.out.println("Names, Emails, Authors, and Titles of Member ID " + memberId + ":");
536       for (String name : member.getNames()) {
537         System.out.println(name);
538       }
539       for (String email : member.getEmails()) {
540         System.out.println(email);
541       }
542       for (String author : member.getAuthors()) {
543         System.out.println(author);
544       }
545       for (String title : member.getTitles()) {
546         System.out.println(title);
547       }
548     } else {
549       System.out.println("Member not found: " + memberId);
550     }
551   }
552
553   public void displayMemberNamesAndEmailsAndTitlesAndAuthors(int memberId) {
554     Member member = members.get(memberId);
555     if (member != null) {
556       System.out.println("Names, Emails, Titles, and Authors of Member ID " + memberId + ":");
557       for (String name : member.getNames()) {
558         System.out.println(name);
559       }
560       for (String email : member.getEmails()) {
561         System.out.println(email);
562       }
563       for (String title : member.getTitles()) {
564         System.out.println(title);
565       }
566       for (String author : member.getAuthors()) {
567         System.out.println(author);
568       }
569     } else {
570       System.out.println("Member not found: " + memberId);
571     }
572   }
573
574   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthors(int memberId) {
575     Member member = members.get(memberId);
576     if (member != null) {
577       System.out.println("Names, Emails, Authors, Titles, and Authors of Member ID " + memberId + ":");
578       for (String name : member.getNames()) {
579         System.out.println(name);
580       }
581       for (String email : member.getEmails()) {
582         System.out.println(email);
583       }
584       for (String title : member.getTitles()) {
585         System.out.println(title);
586       }
587       for (String author : member.getAuthors()) {
588         System.out.println(author);
589       }
590     } else {
591       System.out.println("Member not found: " + memberId);
592     }
593   }
594
595   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitles(int memberId) {
596     Member member = members.get(memberId);
597     if (member != null) {
598       System.out.println("Names, Emails, Authors, Titles, Authors, and Titles of Member ID " + memberId + ":");
599       for (String name : member.getNames()) {
600         System.out.println(name);
601       }
602       for (String email : member.getEmails()) {
603         System.out.println(email);
604       }
605       for (String title : member.getTitles()) {
606         System.out.println(title);
607       }
608       for (String author : member.getAuthors()) {
609         System.out.println(author);
610       }
611     } else {
612       System.out.println("Member not found: " + memberId);
613     }
614   }
615
616   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthors(int memberId) {
617     Member member = members.get(memberId);
618     if (member != null) {
619       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, and Authors of Member ID " + memberId + ":");
620       for (String name : member.getNames()) {
621         System.out.println(name);
622       }
623       for (String email : member.getEmails()) {
624         System.out.println(email);
625       }
626       for (String title : member.getTitles()) {
627         System.out.println(title);
628       }
629       for (String author : member.getAuthors()) {
630         System.out.println(author);
631       }
632     } else {
633       System.out.println("Member not found: " + memberId);
634     }
635   }
636
637   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitles(int memberId) {
638     Member member = members.get(memberId);
639     if (member != null) {
640       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, and Titles of Member ID " + memberId + ":");
641       for (String name : member.getNames()) {
642         System.out.println(name);
643       }
644       for (String email : member.getEmails()) {
645         System.out.println(email);
646       }
647       for (String title : member.getTitles()) {
648         System.out.println(title);
649       }
650       for (String author : member.getAuthors()) {
651         System.out.println(author);
652       }
653     } else {
654       System.out.println("Member not found: " + memberId);
655     }
656   }
657
658   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthors(int memberId) {
659     Member member = members.get(memberId);
660     if (member != null) {
661       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, and Authors of Member ID " + memberId + ":");
662       for (String name : member.getNames()) {
663         System.out.println(name);
664       }
665       for (String email : member.getEmails()) {
666         System.out.println(email);
667       }
668       for (String title : member.getTitles()) {
669         System.out.println(title);
670       }
671       for (String author : member.getAuthors()) {
672         System.out.println(author);
673       }
674     } else {
675       System.out.println("Member not found: " + memberId);
676     }
677   }
678
679   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitles(int memberId) {
680     Member member = members.get(memberId);
681     if (member != null) {
682       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, and Titles of Member ID " + memberId + ":");
683       for (String name : member.getNames()) {
684         System.out.println(name);
685       }
686       for (String email : member.getEmails()) {
687         System.out.println(email);
688       }
689       for (String title : member.getTitles()) {
690         System.out.println(title);
691       }
692       for (String author : member.getAuthors()) {
693         System.out.println(author);
694       }
695     } else {
696       System.out.println("Member not found: " + memberId);
697     }
698   }
699
700   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitles(int memberId) {
701     Member member = members.get(memberId);
702     if (member != null) {
703       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
704       for (String name : member.getNames()) {
705         System.out.println(name);
706       }
707       for (String email : member.getEmails()) {
708         System.out.println(email);
709       }
710       for (String title : member.getTitles()) {
711         System.out.println(title);
712       }
713       for (String author : member.getAuthors()) {
714         System.out.println(author);
715       }
716     } else {
717       System.out.println("Member not found: " + memberId);
718     }
719   }
720
721   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitles(int memberId) {
722     Member member = members.get(memberId);
723     if (member != null) {
724       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, and Titles of Member ID " + memberId + ":");
725       for (String name : member.getNames()) {
726         System.out.println(name);
727       }
728       for (String email : member.getEmails()) {
729         System.out.println(email);
730       }
731       for (String title : member.getTitles()) {
732         System.out.println(title);
733       }
734       for (String author : member.getAuthors()) {
735         System.out.println(author);
736       }
737     } else {
738       System.out.println("Member not found: " + memberId);
739     }
740   }
741
742   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
743     Member member = members.get(memberId);
744     if (member != null) {
745       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
746       for (String name : member.getNames()) {
747         System.out.println(name);
748       }
749       for (String email : member.getEmails()) {
750         System.out.println(email);
751       }
752       for (String title : member.getTitles()) {
753         System.out.println(title);
754       }
755       for (String author : member.getAuthors()) {
756         System.out.println(author);
757       }
758     } else {
759       System.out.println("Member not found: " + memberId);
760     }
761   }
762
763   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
764     Member member = members.get(memberId);
765     if (member != null) {
766       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
767       for (String name : member.getNames()) {
768         System.out.println(name);
769       }
770       for (String email : member.getEmails()) {
771         System.out.println(email);
772       }
773       for (String title : member.getTitles()) {
774         System.out.println(title);
775       }
776       for (String author : member.getAuthors()) {
777         System.out.println(author);
778       }
779     } else {
780       System.out.println("Member not found: " + memberId);
781     }
782   }
783
784   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
785     Member member = members.get(memberId);
786     if (member != null) {
787       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
788       for (String name : member.getNames()) {
789         System.out.println(name);
790       }
791       for (String email : member.getEmails()) {
792         System.out.println(email);
793       }
794       for (String title : member.getTitles()) {
795         System.out.println(title);
796       }
797       for (String author : member.getAuthors()) {
798         System.out.println(author);
799       }
800     } else {
801       System.out.println("Member not found: " + memberId);
802     }
803   }
804
805   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
806     Member member = members.get(memberId);
807     if (member != null) {
808       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
809       for (String name : member.getNames()) {
810         System.out.println(name);
811       }
812       for (String email : member.getEmails()) {
813         System.out.println(email);
814       }
815       for (String title : member.getTitles()) {
816         System.out.println(title);
817       }
818       for (String author : member.getAuthors()) {
819         System.out.println(author);
820       }
821     } else {
822       System.out.println("Member not found: " + memberId);
823     }
824   }
825
826   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
827     Member member = members.get(memberId);
828     if (member != null) {
829       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
830       for (String name : member.getNames()) {
831         System.out.println(name);
832       }
833       for (String email : member.getEmails()) {
834         System.out.println(email);
835       }
836       for (String title : member.getTitles()) {
837         System.out.println(title);
838       }
839       for (String author : member.getAuthors()) {
840         System.out.println(author);
841       }
842     } else {
843       System.out.println("Member not found: " + memberId);
844     }
845   }
846
847   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
848     Member member = members.get(memberId);
849     if (member != null) {
850       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
851       for (String name : member.getNames()) {
852         System.out.println(name);
853       }
854       for (String email : member.getEmails()) {
855         System.out.println(email);
856       }
857       for (String title : member.getTitles()) {
858         System.out.println(title);
859       }
860       for (String author : member.getAuthors()) {
861         System.out.println(author);
862       }
863     } else {
864       System.out.println("Member not found: " + memberId);
865     }
866   }
867
868   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
869     Member member = members.get(memberId);
870     if (member != null) {
871       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
872       for (String name : member.getNames()) {
873         System.out.println(name);
874       }
875       for (String email : member.getEmails()) {
876         System.out.println(email);
877       }
878       for (String title : member.getTitles()) {
879         System.out.println(title);
880       }
881       for (String author : member.getAuthors()) {
882         System.out.println(author);
883       }
884     } else {
885       System.out.println("Member not found: " + memberId);
886     }
887   }
888
889   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
890     Member member = members.get(memberId);
891     if (member != null) {
892       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
893       for (String name : member.getNames()) {
894         System.out.println(name);
895       }
896       for (String email : member.getEmails()) {
897         System.out.println(email);
898       }
899       for (String title : member.getTitles()) {
900         System.out.println(title);
901       }
902       for (String author : member.getAuthors()) {
903         System.out.println(author);
904       }
905     } else {
906       System.out.println("Member not found: " + memberId);
907     }
908   }
909
910   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
911     Member member = members.get(memberId);
912     if (member != null) {
913       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
914       for (String name : member.getNames()) {
915         System.out.println(name);
916       }
917       for (String email : member.getEmails()) {
918         System.out.println(email);
919       }
920       for (String title : member.getTitles()) {
921         System.out.println(title);
922       }
923       for (String author : member.getAuthors()) {
924         System.out.println(author);
925       }
926     } else {
927       System.out.println("Member not found: " + memberId);
928     }
929   }
930
931   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
932     Member member = members.get(memberId);
933     if (member != null) {
934       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
935       for (String name : member.getNames()) {
936         System.out.println(name);
937       }
938       for (String email : member.getEmails()) {
939         System.out.println(email);
940       }
941       for (String title : member.getTitles()) {
942         System.out.println(title);
943       }
944       for (String author : member.getAuthors()) {
945         System.out.println(author);
946       }
947     } else {
948       System.out.println("Member not found: " + memberId);
949     }
950   }
951
952   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
953     Member member = members.get(memberId);
954     if (member != null) {
955       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
956       for (String name : member.getNames()) {
957         System.out.println(name);
958       }
959       for (String email : member.getEmails()) {
960         System.out.println(email);
961       }
962       for (String title : member.getTitles()) {
963         System.out.println(title);
964       }
965       for (String author : member.getAuthors()) {
966         System.out.println(author);
967       }
968     } else {
969       System.out.println("Member not found: " + memberId);
970     }
971   }
972
973   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
974     Member member = members.get(memberId);
975     if (member != null) {
976       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
977       for (String name : member.getNames()) {
978         System.out.println(name);
979       }
980       for (String email : member.getEmails()) {
981         System.out.println(email);
982       }
983       for (String title : member.getTitles()) {
984         System.out.println(title);
985       }
986       for (String author : member.getAuthors()) {
987         System.out.println(author);
988       }
989     } else {
990       System.out.println("Member not found: " + memberId);
991     }
992   }
993
994   public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
995     Member member = members.get(memberId);
996     if (member != null) {
997       System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, and Titles of Member ID " + memberId + ":");
998       for (String name : member.getNames()) {
999         System.out.println(name);
1000      }
1001      for (String email : member.getEmails()) {
1002        System.out.println(email);
1003      }
1004      for (String title : member.getTitles()) {
1005        System.out.println(title);
1006      }
1007      for (String author : member.getAuthors()) {
1008        System.out.println(author);
1009      }
1010    } else {
1011      System.out.println("Member not found: " + memberId);
1012    }
1013  }
1014
1015  public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
1016    Member member = members.get(memberId);
1017    if (member != null) {
1018      System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
1019      for (String name : member.getNames()) {
1020        System.out.println(name);
1021      }
1022      for (String email : member.getEmails()) {
1023        System.out.println(email);
1024      }
1025      for (String title : member.getTitles()) {
1026        System.out.println(title);
1027      }
1028      for (String author : member.getAuthors()) {
1029        System.out.println(author);
1030      }
1031    } else {
1032      System.out.println("Member not found: " + memberId);
1033    }
1034  }
1035
1036  public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
1037    Member member = members.get(memberId);
1038    if (member != null) {
1039      System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
1040      for (String name : member.getNames()) {
1041        System.out.println(name);
1042      }
1043      for (String email : member.getEmails()) {
1044        System.out.println(email);
1045      }
1046      for (String title : member.getTitles()) {
1047        System.out.println(title);
1048      }
1049      for (String author : member.getAuthors()) {
1050        System.out.println(author);
1051      }
1052    } else {
1053      System.out.println("Member not found: " + memberId);
1054    }
1055  }
1056
1057  public void displayMemberNamesAndEmailsAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndAuthorsAndTitlesAndTitlesAndTitlesAndTitlesAndTitlesAndTitles(int memberId) {
1058    Member member = members.get(memberId);
1059    if (member != null) {
1060      System.out.println("Names, Emails, Authors, Titles, Authors, Titles, Authors, Titles, Authors, Titles, Titles, Titles, Titles, and Titles of Member ID " + memberId + ":");
1061      for (String name : member.getNames()) {
1062        System.out.println(name);
1063      }
1064      for (String email : member.getEmails()) {
1065        System.out.println(email);
1066      }
1067      for (String title : member.getTitles()) {
1068        System.out.println(title);
1069      }
1070      for (String author : member.getAuthors()) {
1071       
```

```

Rakesh_Assingment.ipynb ResultManager.java 7 LibrarySystem.java 9+ 
JAVA > LibrarySystem.java > Language Support for Java(TM) by Red Hat > Book > getBookId()
5  public class LibrarySystem {
149    static class LibraryManager {
172      private void saveMembers() {
173        ...
178      } catch (IOException e) {
179        System.out.println("Error saving members: " + e.getMessage());
180      }
181    }
182
183    private void saveToFile() {
184      saveBooks();
185      saveMembers();
186    }
187
188    private void loadFromFile() {
189      try (BufferedReader br = new BufferedReader(new FileReader(BOOK_FILE))) {
190        String line;
191        while ((line = br.readLine()) != null) {
192          Book b = Book.fromString(line);
193          if (b != null) {
194            books.put(b.getBookId(), b);
195          }
196        } catch (IOException e) {
197      }
198
199      try (BufferedReader br = new BufferedReader(new FileReader(MEMBER_FILE))) {
200        String line;
201        while ((line = br.readLine()) != null) {
202          Member m = Member.fromString(line);
203          if (m != null) {
204            members.put(m.getMemberId(), m);
205          }
206        } catch (IOException e) {
207      }
208
209    }
210
211    public void addBook() {
212      try {
213        System.out.print("Enter Book ID: ");
214        int id = sc.nextInt();
215        sc.nextLine();
216        if (books.containsKey(id)) {
217          System.out.println("Book ID already exists.");
218          return;
219        }
220
221      public class LibrarySystem {
222        static class LibraryManager {
223          public void addBook() {
224            System.out.print("Enter Title: ");
225            String title = sc.nextLine();
226            System.out.print("Enter Author: ");
227            String author = sc.nextLine();
228            System.out.print("Enter Category: ");
229            String category = sc.nextLine();
230
231            Book b = new Book(id, title, author, category);
232            books.put(id, b);
233            saveBooks();
234            System.out.println("Book added successfully!");
235          } catch (Exception e) {
236            System.out.println("Error adding book: " + e.getMessage());
237          }
238
239          public void addMember() {
240            try {
241              System.out.print("Enter Member ID: ");
242              int id = sc.nextInt();
243              sc.nextLine();
244              if (members.containsKey(id)) {
245                System.out.println("Member ID already exists.");
246                return;
247              }
248              System.out.print("Enter Name: ");
249              String name = sc.nextLine();
250              System.out.print("Enter Email: ");
251              String email = sc.nextLine();
252
253              Member m = new Member(id, name, email);
254              members.put(id, m);
255              saveMembers();
256              System.out.println("Member added successfully!");
257            } catch (Exception e) {
258              System.out.println("Error adding member: " + e.getMessage());
259            }
260
261          public void issueBook() {
262            try {
263              System.out.print("Enter Member ID: ");

```

```

  Rakesh_Assingment.ipynb  ResultManager.java 7  LibrarySystem.java 9+
JAVA > LibrarySystem.java > Language Support for Java(TM) by Red Hat > Book > getBookId()
5   public class LibrarySystem {
149     static class LibraryManager {
260       public void issueBook() {
261         System.out.print("Enter Book ID: ");
262         int bid = sc.nextInt();
263
264         Book b = books.get(bid);
265         Member m = members.get(mid);
266
267         if (b == null) {
268           System.out.println("Book not found.");
269           return;
270         }
271         if (m == null) {
272           System.out.println("Member not found.");
273           return;
274         }
275         if (b.isIssued()) {
276           System.out.println("Book already issued.");
277           return;
278         }
279
280         b.markAsIssued();
281         m.addIssuedBook(bid);
282         saveToFile();
283         System.out.println("Book issued successfully!");
284       } catch (Exception e) {
285         System.out.println("Error issuing book: " + e.getMessage());
286       }
287     }
288
289     public void returnBook() {
290       try {
291         System.out.print("Enter Member ID: ");
292         int mid = sc.nextInt();
293         System.out.print("Enter Book ID: ");
294         int bid = sc.nextInt();
295
296         Book b = books.get(bid);
297         Member m = members.get(mid);
298
299         if (b == null || m == null) {
300           System.out.println("Invalid IDs.");
301           return;
302         }
303         if (!b.isIssued()) {
304
305           b.markAsReturned();
306           m.returnIssuedBook(bid);
307           saveToFile();
308           System.out.println("Book returned successfully!");
309         } catch (Exception e) {
310           System.out.println("Error returning book: " + e.getMessage());
311         }
312       }
313
314       public void searchBooks() {
315         sc.nextLine();
316         System.out.print("Enter keyword (title/author/category): ");
317         String key = sc.nextLine().toLowerCase();
318         for (Book b : books.values()) {
319           if (b.getTitle().toLowerCase().contains(key)
320               || b.getAuthor().toLowerCase().contains(key)
321               || b.getCategory().toLowerCase().contains(key)) {
322             b.displayBookDetails();
323             System.out.println("-----");
324           }
325         }
326       }
327
328       public void sortBooks() {
329         System.out.println("1. Sort by Title\n2. Sort by Author");
330         int ch = sc.nextInt();
331         List<Book> list = new ArrayList<>(books.values());
332         if (ch == 1) {
333           Collections.sort(list);
334         } else {
335           list.sort(Comparator.comparing(Book::getAuthor));
336         }
337         for (Book b : list) {
338           b.displayBookDetails();
339           System.out.println("-----");
340         }
341       }
342
343       public void menu() {
344         while (true) {
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
533
534
535
536
537
537
538
539
539
540
541
542
542
543
544
545
545
546
547
547
548
549
549
550
551
551
552
553
553
554
555
555
556
557
557
558
559
559
560
561
561
562
563
563
564
565
565
566
567
567
568
569
569
570
571
571
572
573
573
574
575
575
576
577
577
578
579
579
580
581
581
582
583
583
584
585
585
586
587
587
588
589
589
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1
```

```
⌚ Rakesh_Assingment.ipynb    ⌚ ResultManager.java 7    ⌚ LibrarySystem.java 9+ X
JAVA > ⌚ LibrarySystem.java > Language Support for Java(TM) by Red Hat > ⌚ LibrarySystem > ⌚ Book > ⌚ getBookId()
5   public class LibrarySystem {
149     static class LibraryManager {
349       public void menu() {
351         System.out.println("\n===== City Library Digital Management System =====");
352         System.out.println("1. Add Book");
353         System.out.println("2. Add Member");
354         System.out.println("3. Issue Book");
355         System.out.println("4. Return Book");
356         System.out.println("5. Search Books");
357         System.out.println("6. Sort Books");
358         System.out.println("7. Exit");
359         System.out.print("Enter choice: ");
360
361         int ch = sc.nextInt();
362         switch (ch) {
363           case 1 ->
364             addBook();
365           case 2 ->
366             addMember();
367           case 3 ->
368             issueBook();
369           case 4 ->
370             returnBook();
371           case 5 ->
372             searchBooks();
373           case 6 ->
374             sortBooks();
375           case 7 -> {
376             saveToFile();
377             System.out.println("Exiting program. Data saved.");
378             return;
379           }
380           default ->
381             System.out.println("Invalid choice!");
382         }
383       }
384     }
385   }
386
Run main | Debug main | Run | Debug
387   public static void main(String[] args) {
388     LibraryManager manager = new LibraryManager();
389     manager.menu();
390   }
391 }
392 }
```

# Output:

```
PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER 11

===== City Library Digital Management System =====
1. Add Book
2. Add Member
3. Issue Book
4. Return Book
5. Search Books
6. Sort Books
7. Exit
Enter choice: 1
Enter Book ID: 01
Enter Title: Java
Enter Author: Rakesh
Enter Category: Coding
Book added successfully!

===== City Library Digital Management System =====
1. Add Book
2. Add Member
3. Issue Book
4. Return Book
5. Search Books
6. Sort Books
7. Exit
Enter choice: 2
Enter Member ID: 02
Enter Name: Tina
Enter Email: 64@gmail.com
Member added successfully!

===== City Library Digital Management System =====
1. Add Book
2. Add Member
3. Issue Book
4. Return Book
5. Search Books
6. Sort Books
7. Exit
Enter choice: 7
Exiting program. Data saved.

d:\RAKESH\VSC\JAVA>
```

# Explanation

## **Objective:**

To develop a simple digital library management system using **File Handling** and **Java Collections Framework (JCF)**.

It helps to **add, issue, return, search, and sort books**, while maintaining **member details**.

## **Key Concepts Used:**

### **1. Classes & Objects:**

- Book → Stores book details (ID, title, author, category, status).
- Member → Stores member details (ID, name, email, issued books).
- LibraryManager → Performs operations like add, issue, return, search, and sort.

### **2. File Handling:**

- Uses FileReader, FileWriter, BufferedReader, and BufferedWriter to **save and load data** from books.txt and members.txt.
- Ensures that data is **persistent** even after the program closes.

### **3. Java Collections Framework (JCF):**

- HashMap → Stores books and members using their IDs as keys.
- ArrayList → Used for managing lists of issued books.

### **4. Interfaces:**

- Comparable<Book> → Enables sorting books by **title**.
- Comparator → Used for sorting by **author**.

### **5. Exception Handling:**

- Handles errors like **invalid input, file not found, or duplicate IDs** using try-catch blocks.