Daily practice-a.

1) Using Arraylist.

create Book class;

id, title, author, availability

Array list of Book.

void display menu

if choice = Add

id, title, author

Add new book

void search book

if (book.title = title)

(book).

void remove book(id).

if (book.bookid = id)

(remove book)

Output:

Book Added

101 Java James

Book found

Book removed

2. Linked list.

use linked list package

create class Book.

id, title, borrowerName,

static LinkedList<Book> issued Books = new LinkedList

void issue book.

id, title, borrower

Add book data to issuedBooks.
Print "Book issued"
Display issued books
Remove book 101
print "book returned"
stop

Output:
Book issued
101 Java Ravi
Book returned

3. Hash set

Create (HashSet<Book> bookSet
Add Book(101, "Java", "James")
Add Book(101, "Java", "James")
FOR each book in bookSet
    print details
END for
Stop

Output:
Book Added
Duplicated not allowed
101 Java James.

4. Tree set:

Create TreeSet<Book> bookTree
Add Book("Python")
Add Book("Java")
Add Book("C")

Display bookTree (sorted)

Stop

Output:

C

Java

python

5. Hashmap:

Start

create Hashmap<Integer, book> bookMap

put (101, Java)

put (102, python)

Search key 101

print result

Remove key 101

Display all books

Stop

Output:

Book found: Java

Book Removed

102 python.

6. Iterator:

Start

Create Arraylist <Book> booklist

Add 2 books

Create Iterator

while hasNext

print book

END while

Use iterator.remove() for id 101

Stop

Output:

101 Java

102 Python

Book Removed

7. Multiple collection:

Start

create Arraylist books

create Hashset authors

Add Book(101, Java, James)

Add author to authorSet

Add Book(102, Python, John)

Add author to authorSet

Display authors

STOP

Output:

James

John

8. Treemap:

Start

Create TreeMap<Integer, Books>

put(102, python)

put(101, Java)

Display TreeMap (sorted by key)

STOP

Output:

101 Java

102 Python

9. LinkedList+HashMap

Start

Create Hashmap allbooks

Create Linkedlist issuedbooks

Add books to allbooks.

Issue book 01 -> add to issuedbooks

Display issuedbooks

Display availablebooks

Stop.

Output:

Issued: 101 Java

Available: 102 python

10. Array Index out of boundsexception

Start

Create book array size 2.

Try

Add 3 books.

Catch ArrayIndexOutOfBoundsException

print "Array Full".

End Try

Stop

Output:

Book added

Book added

Array full

11. Null pointerexception:

```
Start:
Book book=null
Try
   print book.title
catch
   Null pointerexception
  print "Book not found"
stop
```

Output:

Book not found.

12. Class not Foundexception:

```
Start
try
  class.forname('Book')
catch classnotfoundexception
  print "class not found"
stop
```

Output:

Class not found.

13. IOException

```
Start
  Try
    Open file "books.txt"
Catch
     IOException
  print "File Error"
STOP
```

Output:

File error.

14 Multiple catch.

```
Start
Try
  Read integer input
Catch
  input MismatchException
  print "Invalid Input"
Catch
  NullPointerException
  print "Null Error"
Catch
  IOException
  print "IO Error"
STOP
```

Output:

Invalid input

15. Throws keyword:

```
function readfile() throws IOException
  Read file
MAIN
 TRY
   readfile()
 catch
   IO Exception
   Print "Exception handled"
Stop
```

Output:

Exception Handled

16. Custom exception:

Start

Try

If book not found

Throw BookNotFoundException

catch BookNotFoundException

print "Book not available".

Stop

Output:

Book Not Available.

17. Finally block:

Start

Try

open file

catch exception

print "Error"

finally

close file

print 'file closed'

stop

Output:

error

file closed

18. Exception during issue:

start

if book already issued

Throw Exception

catch Exception

print "Book Already Issued"

Stop

Output:

Book Already issued

Main.java

Share Run

```
80      public static void main(String[] args) {
81
82          while (true) {
83              System.out.println("1. Add Book");
84              System.out.println("2. Issue Book");
85              System.out.println("3. Return Book");
86              System.out.println("4. Display Available Books");
87              System.out.println("5. Display Issued Books");
88              System.out.println("6. Exit");
89              int ch = sc.nextInt();
90              sc.nextLine();
91
92              switch (ch) {
93                  case 1 -> addBook();
94                  case 2 -> issueBook();
95                  case 3 -> returnBook();
96                  case 4 -> displayAvailable();
97                  case 5 -> displayIssued();
98                  case 6 -> System.exit(0);
99              }
100         }
```

Output Clear

```
1. Add Book
2. Issue Book
3. Return Book
4. Display Available Books
5. Display Issued Books
6. Exit
4
Available Books:

1. Add Book
2. Issue Book
3. Return Book
4. Display Available Books
5. Display Issued Books
6. Exit
```

Main.java

Share Run

```
8
9           BufferedReader br = null;
10
11          try {
12              br = new BufferedReader(new FileReader("books.txt"
                    ));
13              String line;
14
15              while ((line = br.readLine()) != null) {
16                  System.out.println(line);
17              }
18          } catch (IOException e) {
19              System.out.println("File error occurred.");
20          } finally {
21              try {
22                  if (br != null)
23                      br.close();
24              } catch (IOException e) {
25                  System.out.println("Error closing file.");
26              }
27          }
28      }
29  }
```

Main.java

Share Run

```
        String title = sc.nextLine();

        books[index] = new Book(id, title);
    }

    static void searchBook(int index) {
        try {
            books[index].display();
        } catch (NullPointerException e) {
            System.out.println("Book not found.\n");
        }
    }

    public static void main(String[] args) {

        addBook(0);

        System.out.print("Enter index to search: ");
        int index = sc.nextInt();

        searchBook(index);
    }
}
```

Output

Clear

```
Enter Book ID: 33
Enter Title: tears
Enter index to search: 2
Book not found.


=== Code Execution Successful ===
```

Main.java

Share Run

```java
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

class Book {
    int bookId;
    String title;
    String author;

    Book(int bookId, String title, String author) {
        this.bookId = bookId;
        this.title = title;
        this.author = author;
    }

    void display() {
        System.out.println("ID: " + bookId + ", Title: " +
            title + ", Author: " + author);
    }
}

public class Main {
```

Output Clear

```
---- Library Management System ----
1. Add Book
2. Display Books
3. Remove Book
4. Exit
Enter your choice: 1
Enter Book ID: 25
Enter Title: tiger
Enter Author: saniya
Book added successfully.

---- Library Management System ----
1. Add Book
2. Display Books
3. Remove Book
4. Exit
Enter your choice:
```

Main.java

Share Run

```
88          switch (choice) {
89              case 1:
90                  addBook();
91                  break;
92              case 2:
93                  searchBook();
94                  break;
95              case 3:
96                  removeBook();
97                  break;
98              case 4:
99                  displayBooks();
100                 break;
101             case 5:
102                 System.out.println("Exiting...");
103                 System.exit(0);
104             default:
105                 System.out.println("Invalid choice.\n");
106         }
107     }
108 }
109 }
110
```

Output

Clear

```
---- Library Management System ----
1. Add Book
2. Search Book by ID
3. Remove Book
4. Display All Books
5. Exit
Enter your choice: 2
Enter Book ID to search: 32
Book not found.

---- Library Management System ----
1. Add Book
2. Search Book by ID
3. Remove Book
4. Display All Books
5. Exit
Enter your choice:
```

Main.java

Share

Run

Output

Clear

```java
public class Main {

    public static void main(String[] args) {

        try {
            Class.forName("Book");
            System.out.println("Book class loaded successfully."
                );
        } catch (ClassNotFoundException e) {
            System.out.println("Book class not found.");
        }
    }
}
```

```
Book class not found.

=== Code Execution Successful ===
```

Main.java

Share Run

```
62          System.out.println("3. Exit");
63          System.out.print("Enter your choice: ");
64
65          int choice = sc.nextInt();
66          sc.nextLine();
67
68          switch (choice) {
69              case 1:
70                  addBook();
71                  break;
72              case 2:
73                  displayBooks();
74                  break;
75              case 3:
76                  System.out.println("Exiting...");
77                  System.exit(0);
78              default:
79                  System.out.println("Invalid choice.\n");
80          }
81      }
82  }
83 }
84
```

Output Clear

```
---- Library Management System ----
1. Add Book
2. Display Books (Sorted by Title)
3. Exit
Enter your choice: 2
No books available.

---- Library Management System ----
1. Add Book
2. Display Books (Sorted by Title)
3. Exit
Enter your choice:
```

Main.java

Share

Run

```
import java.io.FileReader;
import java.io.IOException;

public class Main {

    public static void main(String[] args) {

        try {
            BufferedReader br = new BufferedReader(new
                FileReader("books.txt"));
            String line;

            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }

            br.close();
        } catch (IOException e) {
            System.out.println("File missing or unreadable.");
        }
    }
}
```

Main.java

Share Run

```
27            if (b.bookId == id) {
28                System.out.println("Book Found: " + b.title);
29                return;
30            }
31        }
32        throw new BookNotFoundException("Book not found.");
33    }
34
35    public static void main(String[] args) {
36
37        library.add(new Book(101, "Java Programming"));
38        library.add(new Book(102, "Python Basics"));
39
40        System.out.print("Enter Book ID to search: ");
41        int id = sc.nextInt();
42
43        try {
44            searchBook(id);
45        } catch (BookNotFoundException e) {
46            System.out.println(e.getMessage());
47        }
48    }
49 }
```

Main.java

Share Run

```
63
64      public static void main(String[] args) {
65
66          while (true) {
67              System.out.println("1. Add Book");
68              System.out.println("2. Delete Book");
69              System.out.println("3. Search Book");
70              System.out.println("4. Display All Books");
71              System.out.println("5. Exit");
72              int ch = sc.nextInt();
73              sc.nextLine();
74
75              switch (ch) {
76                  case 1 -> addBook();
77                  case 2 -> deleteBook();
78                  case 3 -> searchBook();
79                  case 4 -> displayBooks();
80                  case 5 -> System.exit(0);
81              }
82          }
83      }
84  }
85
```

Output

Clear

```
1. Add Book
2. Delete Book
3. Search Book
4. Display All Books
5. Exit
2
Enter Book ID: 24
Book deleted.

1. Add Book
2. Delete Book
3. Search Book
4. Display All Books
5. Exit
```

Main.java

Share Run

```
25            int id = sc.nextInt();
26            sc.nextLine();
27
28            System.out.print("Enter Title: ");
29            String title = sc.nextLine();
30
31            book = new Book(id, title);
32
33            FileWriter fw = new FileWriter("bookdata.txt");
34            fw.write(book.bookId + " " + book.title);
35            fw.close();
36
37            System.out.println("Book saved.");
38        } catch (InputMismatchException e) {
39            System.out.println("Invalid input type.");
40        } catch (NullPointerException e) {
41            System.out.println("Book object is null.");
42        } catch (IOException e) {
43            System.out.println("File error occurred.");
44        }
45    }
46 }
47
```

Main.java

Share Run

```
31  }
32
33  public class Main {
34
35      public static void main(String[] args) {
36
37          HashSet<Book> library = new HashSet<>();
38
39          library.add(new Book(101, "Java Programming", "James
                Gosling"));
40          library.add(new Book(102, "Python Basics", "Guido van
                Rossum"));
41          library.add(new Book(101, "Java Programming", "James
                Gosling"));
42
43          System.out.println("Available Books:");
44
45          for (Book b : library) {
46              b.display();
47          }
48      }
49  }
50
```

Output

Clea

```
Available Books:
ID: 101, Title: Java Programming, Author: James Gosling
ID: 102, Title: Python Basics, Author: Guido van Rossum

=== Code Execution Successful ===
```

Main.java

Share Run

```
15      static void issueBook(int id) throws
            BookAlreadyIssuedException {
16          if (issuedBooks.contains(id)) {
17              throw new BookAlreadyIssuedException("Book already
                    issued.");
18          }
19          issuedBooks.add(id);
20          System.out.println("Book issued successfully.");
21      }
22
23      public static void main(String[] args) {
24
25          System.out.print("Enter Book ID to issue: ");
26          int id = sc.nextInt();
27
28          try {
29              issueBook(id);
30          } catch (BookAlreadyIssuedException e) {
31              System.out.println(e.getMessage());
32          }
33      }
34  }
35
```

Main.java

Share Run

```
    static void readBooks() throws IOException {
        BufferedReader br = new BufferedReader(new FileReader
            ("books.txt"));
        String line;

        while ((line = br.readLine()) != null) {
            System.out.println(line);
        }

        br.close();
    }

    public static void main(String[] args) {

        try {
            readBooks();
        } catch (IOException e) {
            System.out.println("Error reading file.");
        }
    }
}
```

Output Clear

```
Error reading file.

=== Code Execution Successful ===
```

Main.java

Share Run

```
            break;
        }
    }

    if (!removed) {
        System.out.println("Book ID not found.\n");
    }
}

public static void main(String[] args) {

    while (true) {
        System.out.println("---- Library Management System
            ----");
        System.out.println("1. Add Book");
        System.out.println("2. Display All Books");
        System.out.println("3. Search Book by Title");
        System.out.println("4. Remove Book by ID");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");

        int choice = sc.nextInt();
```

Output Clear

```
---- Library Management System ----
1. Add Book
2. Display All Books
3. Search Book by Title
4. Remove Book by ID
5. Exit
Enter your choice: 3
Enter title to search: 5
Book not found.

---- Library Management System ----
1. Add Book
2. Display All Books
3. Search Book by Title
4. Remove Book by ID
5. Exit
Enter your choice:
```

Main.java

Share Run

```
            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    issueBook();
                    break;
                case 2:
                    returnBook();
                    break;
                case 3:
                    displayIssuedBooks();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice.\n");
            }
        }
    }
}
```

Output

Clear

```
---- Library Management System ----
1. Issue Book
2. Return Book
3. Display Issued Books
4. Exit
Enter your choice: 4
Exiting...

=== Code Execution Successful ===
```

Main.java

Share Run

```
        for (int i = 0; i < count; i++) {
            books[i].display();
        }
        System.out.println();
    }

    public static void main(String[] args) {

        while (true) {
            System.out.println("1. Add Book");
            System.out.println("2. Display Books");
            System.out.println("3. Exit");
            int ch = sc.nextInt();
            sc.nextLine();

            switch (ch) {
                case 1 -> addBook();
                case 2 -> displayBooks();
                case 3 -> System.exit(0);
            }
        }
    }
}
```

Output Clear

```
1. Add Book
2. Display Books
3. Exit
```

Main.java

Share Run

```
55          System.out.println();
56      }
57
58      public static void main(String[] args) {
59
60          while (true) {
61              System.out.println("1. Add Book");
62              System.out.println("2. Display Books");
63              System.out.println("3. Display Authors");
64              System.out.println("4. Exit");
65              int ch = sc.nextInt();
66              sc.nextLine();
67
68              switch (ch) {
69                  case 1 -> addBook();
70                  case 2 -> displayBooks();
71                  case 3 -> displayAuthors();
72                  case 4 -> System.exit(0);
73              }
74          }
75      }
76  }
77
```

Output

Clear

```
1. Add Book
2. Display Books
3. Display Authors
4. Exit
3
Authors:

1. Add Book
2. Display Books
3. Display Authors
4. Exit
```