Java Lab Assignment 4

A Java Application for a basic shape drawing application

Problem Statement

Implement a **Student Record Management System** with persistent storage using **file handling** and **Java Collections Framework**. The system should read student records from a file (students.txt) at the start of the application and save updated records back to the file upon exit. The records should be managed using collections like **ArrayList** or **HashMap** to store student information, and should be **sorted** by marks using **Comparator**. The system should allow for viewing, sorting, and displaying student data using **Iterator**. Additionally, implement file attributes using the **File** class and demonstrate reading records randomly using **RandomAccessFile**.

Objective:
Implement file handling and use the collections API to manage student records efficiently.

Learning Outcomes

Upon completion of this assignment, the student will be able to:

1. Implement **file handling** for storing and retrieving student records.
2. Use **collections** (List, Map) to manage and manipulate records.
3. Sort and display records using **Comparator**, **Comparable**, and **Iterator**.

Class Hierarchy & Data Types

Class Hierarchy:

1. **FileUtil**: Contains methods for reading and writing to file.
2. **StudentManager**: Manages student records.

Data Types:

- ArrayList<Student>: For managing student records.
- BufferedReader, BufferedWriter: For file handling.

Detailed Instructions

1. **File Handling**: Use **BufferedReader** and **BufferedWriter** to read and write student data to a file.
2. **Sorting**: Sort students by marks using **Comparator**.
3. **Displaying**: Use **Iterator** to display student data.

Expected Output

```
Loaded students from file:
Roll No: 101
Name: Ankit
Email: ankit@mail.com
Course: B.Tech
Marks: 85.5
--------------------
Roll No: 102
Name: Riya
Email: riya@mail.com
Course: M.Tech
Marks: 91.0
--------------------

===== Capstone Student Menu =====
1. Add Student
2. View All Students
3. Search by Name
4. Delete by Name
5. Sort by Marks
6. Save and Exit
Enter choice: 1
Enter Roll No: 103
Enter Name: Karan
Enter Email: karan@mail.com
Enter Course: BCA
Enter Marks: 76.2

Sorted Student List by Marks:
Roll No: 102
Name: Riya
Email: riya@mail.com
Course: M.Tech
Marks: 91.0
--------------------
```

Guidelines to Students

1. **File Handling**: Ensure proper error handling for file read/write operations.
2. **Collections**: Use **Map** and **List** for managing and displaying student data.
3. **Sorting**: Implement sorting both by marks and by name using **Comparator**.

Improvements/Adjustments

1. **Queue Implementation**: Use **PriorityQueue** for storing and retrieving students based on different criteria.
2. **File Format Enhancement**: Consider saving records in a more structured format like **CSV** or **JSON**.

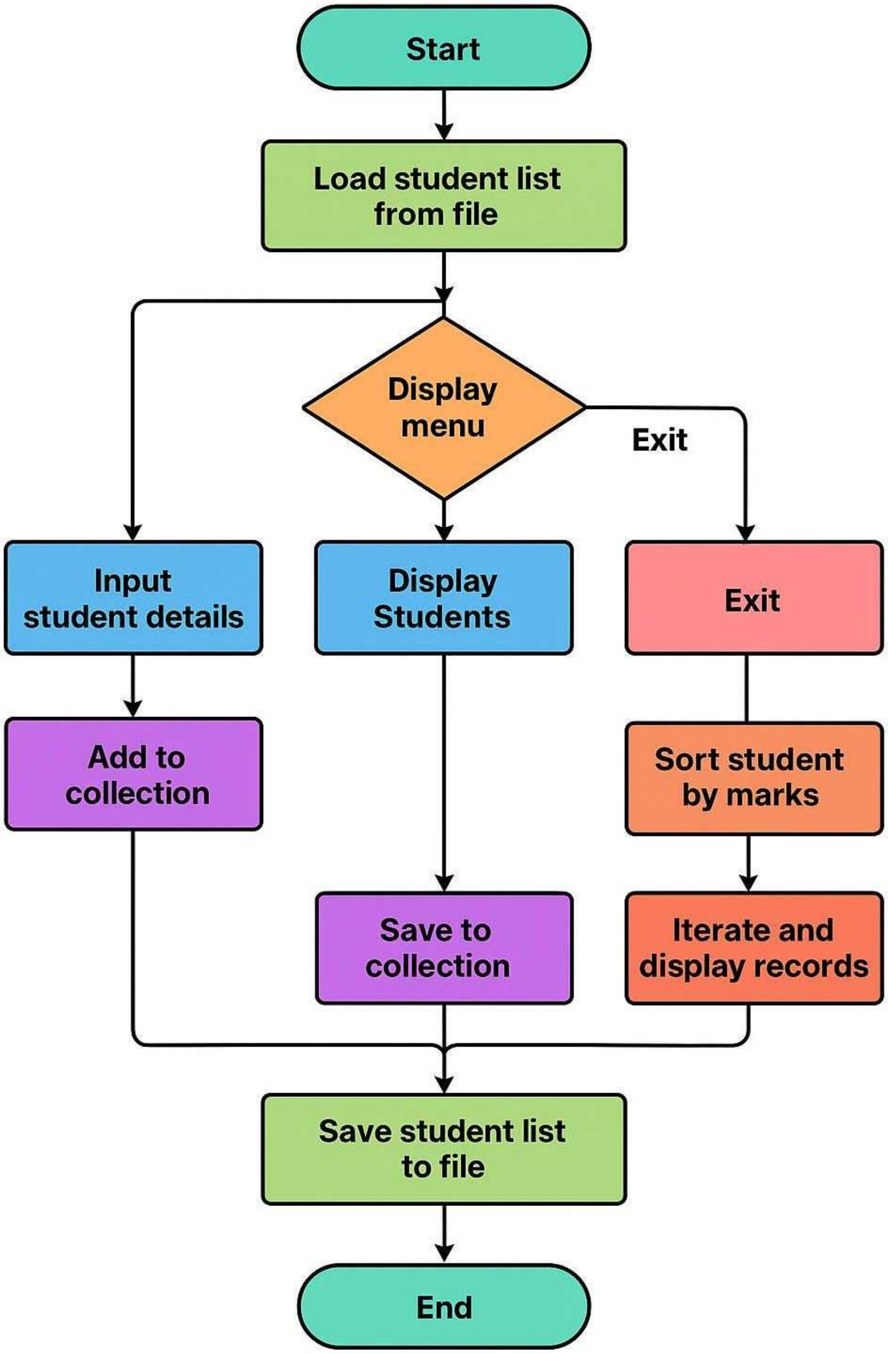
Submission Guidelines

1. Submit **all source files** along with necessary libraries.
2. Ensure proper formatting and comments in your code.

Performance Metrics (Out of 10 Marks)

Criteria	**Marks**
File Handling and Persistence	3
Sorting and Display with Collections	2
Iterator and Data Management	2
Code Structure and Comments	2
Testing and Output Validation	1

Flow Chart:


Start
Load student list from file
Display menu
Exit
Input student details
Display Students
Exit
Add to collection
Sort student by marks
Save to collection
Iterate and display records
Save student list to file
End