



K.R. Mangalam University  
School of Engineering & Technology

Fundamentals Of Java Programming Lab  
(ENCA203) Assignment 4  
Student 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)

## 1. Student.java

```
FileUtil.java  ComparatorByMarks.java 1  Main.java 1  ResultManager.java 7  Student.java 6  StudentManager.java
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > Student.java > ...
1  package StudentRecordSystem;
2
3  public class Student          // RAKESH G    2401201064
4  {
5      private int rollNo;      Field rollNo can be final
6      private String name;     Field name can be final
7      private String email;    Field email can be final
8      private String course;   Field course can be final
9      private double marks;    Field marks can be final
10
11     public Student(int rollNo, String name, String email, String course, double marks)
12     {
13         this.rollNo = rollNo;
14         this.name = name;
15         this.email = email;
16         this.course = course;
17         this.marks = marks;
18     }
19     public int getRollNo() { return rollNo; }
20     public String getName() { return name; }
21     public String getEmail() { return email; }
22     public String getCourse() { return course; }
23     public double getMarks() { return marks; }
24
25     public String toString()    Add @Override Annotation
26     {
27         return "Roll No: " + rollNo +
28             "\nName: " + name +
29             "\nEmail: " + email +
30             "\nCourse: " + course +
31             "\nMarks: " + marks + "\n";
32     }
33     public String toCSV()
34     {
35         return rollNo + "," + name + "," + email + "," + course + "," + marks;
36     }
37 }
38
```

## 2. ComparatorByMarks.java

```
FileUtil.java  ComparatorByMarks.java 1  Main.java 1  ResultManager.java 7  Student.java 6  StudentManager.java
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > ComparatorByMarks.java > ...
1  package StudentRecordSystem;
2  import java.util.Comparator;    // RAKESH G    2401201064
3
4  public class ComparatorByMarks implements Comparator<Student>
5  {
6      public int compare(Student s1, Student s2)    Add @Override Annotation
7      {
8          return Double.compare(s2.getMarks(), s1.getMarks()); // Descending order
9      }
10 }
11
12
13
```

### 3. FileUtil.java

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > FileUtil.java > Java > FileUtil
1 package StudentRecordSystem;
2 import java.io.*;
3 import java.util.*;
4 public class FileUtil // RAKESH G 2401201064
5 {
6     private static final String FILE_NAME = "students.txt";
7     public static List<Student> readStudentsFromFile()
8     {
9         List<Student> list = new ArrayList<>();
10        try (BufferedReader br = new BufferedReader(new FileReader(FILE_NAME)))
11        {
12            String line;
13            while ((line = br.readLine()) != null)
14            {
15                String[] parts = line.split(",");
16                if (parts.length == 5)
17                {
18                    int rollNo = Integer.parseInt(parts[0]);
19                    String name = parts[1];
20                    String email = parts[2];
21                    String course = parts[3];
22                    double marks = Double.parseDouble(parts[4]);
23                    list.add(new Student(rollNo, name, email, course, marks));
24                }
25            }
26        }
27        catch (IOException e)
28        {
29            System.out.println("File not found or error reading file. A new file will be created on save.");
30        }
31        return list;
32    }
33    public static void writeStudentsToFile(List<Student> students)
34    {
35        try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE_NAME)))
36        {
37            for (Student s : students)
38                bw.write(s.toCSV() + "\n");
39        }
40    }
41    catch (IOException e)
42    {
43        System.out.println("Error writing to file: " + e.getMessage());
44    }
45    }
46    public static void showFileAttributes()
47    {
48        File file = new File(FILE_NAME);
49        System.out.println("\n--- File Attributes ---");
50        System.out.println("File Name: " + file.getName());
51        System.out.println("Path: " + file.getAbsolutePath());
52        System.out.println("Size: " + file.length() + " bytes");
53        System.out.println("Readable: " + file.canRead());
54        System.out.println("Writable: " + file.canWrite());
55        System.out.println("-----\n");
56    }
57    public static void readRandomRecord()
58    {
59        try (RandomAccessFile raf = new RandomAccessFile("students.txt", "r"))
60        {
61            long length = raf.length();
62            long position = length / 2;
63            raf.seek(position);
64            String line = raf.readLine();
65            System.out.println("Random Record Read:\n" + line);
66        }
67        catch (IOException e)
68        {
69            System.out.println("Error reading random record: " + e.getMessage());
70        }
71    }
72    }
73 }
```

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > FileUtil.java > Java > FileUtil
4 public class FileUtil // RAKESH G 2401201064
33 public static void writeStudentsToFile(List<Student> students)
38 {
39     bw.write(s.toCSV() + "\n");
40     System.out.println("Data saved successfully to " + FILE_NAME);
41 }
42 catch (IOException e)
43 {
44     System.out.println("Error writing to file: " + e.getMessage());
45 }
46 }
47 public static void showFileAttributes()
48 {
49     File file = new File(FILE_NAME);
50     System.out.println("\n--- File Attributes ---");
51     System.out.println("File Name: " + file.getName());
52     System.out.println("Path: " + file.getAbsolutePath());
53     System.out.println("Size: " + file.length() + " bytes");
54     System.out.println("Readable: " + file.canRead());
55     System.out.println("Writable: " + file.canWrite());
56     System.out.println("-----\n");
57 }
58 public static void readRandomRecord()
59 {
60     try (RandomAccessFile raf = new RandomAccessFile("students.txt", "r"))
61     {
62         long length = raf.length();
63         long position = length / 2;
64         raf.seek(position);
65         String line = raf.readLine();
66         System.out.println("Random Record Read:\n" + line);
67     }
68     catch (IOException e)
69     {
70         System.out.println("Error reading random record: " + e.getMessage());
71     }
72 }
73 }
```

## 4. StudentManager.java

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > StudentManager.java > ...
1 package StudentRecordSystem;
2
3 import java.util.*; // RAKESH G 2401201064
4
5 public class StudentManager
6 {
7     private List<Student> students;
8
9     public StudentManager()
10    {
11        students = new ArrayList<>();
12    }
13
14    public void loadStudents()
15    {
16        students = FileUtil.readStudentsFromFile();
17        System.out.println("Loaded students from file:");
18        viewAllStudents();
19        FileUtil.showFileAttributes();
20        FileUtil.readRandomRecord();
21    }
22
23    public void saveStudents()
24    {
25        FileUtil.writeStudentsToFile(students);
26    }
27
28    public void addStudent()
29    {
30        Scanner sc = new Scanner(System.in);
31        System.out.print("Enter Roll No: ");
32        int roll = sc.nextInt();
33        sc.nextLine();
34        System.out.print("Enter Name: ");
35        String name = sc.nextLine();
36        System.out.print("Enter Email: ");
37        String email = sc.nextLine();
38        System.out.print("Enter Course: ");
39        String course = sc.nextLine();
40        System.out.print("Enter Marks: ");
41        double marks = sc.nextDouble();
42
43        students.add(new Student(roll, name, email, course, marks));
44        System.out.println("Student added successfully!\n");
45    }
46
```

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > StudentManager.java > Java > StudentManager > deleteByName(String name)
5 public class StudentManager
46
47     public void viewAllStudents()
48     {
49         Iterator<Student> it = students.iterator();
50         while (it.hasNext())
51         {
52             System.out.println(it.next());
53         }
54     }
55     public void searchByName(String name)
56     {
57         for (Student s : students)
58         {
59             if (s.getName().equalsIgnoreCase(name))
60             {
61                 System.out.println("Student Found:\n" + s);
62                 return;
63             }
64         }
65         System.out.println("No student found with name: " + name);
66     }
67     public void deleteByName(String name)
68     {
69         Iterator<Student> it = students.iterator();
70         boolean removed = false;
71         while (it.hasNext())
72         {
73             if (it.next().getName().equalsIgnoreCase(name))
74             {
75                 it.remove();
76                 removed = true;
77             }
78         }
79         if (removed)
80             System.out.println("Student deleted successfully!");
81         else
82             System.out.println("No student found with that name.");
83     }
84     public void sortByMarks()
85     {
86         students.sort(new ComparatorByMarks());
87         System.out.println("Sorted Student List by Marks:");
88         viewAllStudents();
89     }
90 }
```

## 5. Main.java

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > Main.java > Java > Main > main(String[] args)
1  package StudentRecordSystem;
2  import java.util.*;
3  public class Main // RAKESH G 2401201064
4  {
5      public static void main(String[] args)
6      {
7          StudentManager manager = new StudentManager();
8          manager.loadStudents();
9
10         Scanner sc = new Scanner(System.in);
11         int choice;
12
13         do
14         {
15             System.out.println("==== Capstone Student Menu =====");
16             System.out.println("1. Add Student");
17             System.out.println("2. View All Students");
18             System.out.println("3. Search by Name");
19             System.out.println("4. Delete by Name");
20             System.out.println("5. Sort by Marks");
21             System.out.println("6. Save and Exit");
22             System.out.print("Enter choice: ");
23             choice = sc.nextInt();
24             sc.nextLine();
25
26             switch (choice) Convert switch to rule switch
27             {
28                 case 1:
29                     manager.addStudent();
30                     break;
31
32                 case 2:
33                     manager.viewAllStudents();
34                     break;
35
36                 case 3:
37                     System.out.print("Enter name to search: ");
```

```
FileUtil.java • ComparatorByMarks.java 1 • Main.java 1 • ResultManager.java 7 • Student.java 6 • StudentManager.java •
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > Main.java > Java > Main > main(String[] args)
3  public class Main // RAKESH G 2401201064
5      public static void main(String[] args)
29         manager.addStudent();
30         break;
31
32         case 2:
33             manager.viewAllStudents();
34             break;
35
36         case 3:
37             System.out.print("Enter name to search: ");
38             String searchName = sc.nextLine();
39             manager.searchByName(searchName);
40             break;
41
42         case 4:
43             System.out.print("Enter name to delete: ");
44             String delName = sc.nextLine();
45             manager.deleteByName(delName);
46             break;
47
48         case 5:
49             manager.sortByMarks();
50             break;
51
52         case 6:
53             manager.saveStudents();
54             System.out.println("Records saved. Exiting...");
55             break;
56
57         default:
58             System.out.println("Invalid choice! Try again.");
59     }
60 } while (choice != 6);
61 }
62 }
63 }
64 }
```

## 6. Output

```
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER 6

Active code page: 65001

D:\RAKESH\VSC>cd "d:\RAKESH\VSC\StudentRecordSystem (2) java lab 4\StudentRecordSystem\" && javac ResultManager.java && java ResultManager
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8

===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 1
Enter Roll Number: 64
Enter Student Name: Rakesh
Enter marks for subject 1: 98
Enter marks for subject 2: 97
Enter marks for subject 3: 95
Student added successfully. Returning to main menu...

===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 2
Enter Roll Number to search: 64
Roll Number: 64
Student Name: Rakesh
Marks: 98 97 95
Average: 96.66666666666667
Result: Pass
Search completed.

===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 3
Exiting program. Thank you!
Scanner closed. Program terminated.

d:\RAKESH\VSC\StudentRecordSystem (2) java lab 4\StudentRecordSystem>
```

## **Explanation:**

The **Student Record Management System** is a Java-based application developed to manage student details efficiently using **file handling** and the **Java Collections Framework**. The main objective of this program is to store, retrieve, and manipulate student information such as roll number, name, email, course, and marks in a structured and organized way.

At the beginning of the program, student records are **loaded from an external file (students.txt)** using **BufferedReader**, ensuring data persistence even after the program ends. The program allows users to perform various operations such as **adding a new student, viewing all students, searching by name, deleting a record, and sorting students by marks**. When the user exits, all updated data is **saved back to the file** using **BufferedWriter**.

The system uses an **ArrayList** to store multiple student objects dynamically. It applies the **Comparator** interface to sort student records based on marks and uses the **Iterator** interface to traverse and display data efficiently. Additionally, file attributes are accessed through the **File** class, and random record reading is demonstrated using the **RandomAccessFile** class.

The project is divided into several Java classes:

- **Student.java**: Defines the student entity and its attributes.
- **ComparatorByMarks.java**: Sorts students in descending order of marks.
- **FileUtil.java**: Handles reading, writing, and file-related operations.
- **StudentManager.java**: Manages operations such as add, search, delete, and sort.
- **Main.java**: Provides a menu-driven interface for user interaction.

This assignment helped in understanding how **Object-Oriented Programming (OOP)**, **Collections**, and **File Handling** can be integrated to build a complete data management application. It demonstrates the use of **Encapsulation, Iterators, Comparators, and Persistent Storage**, making it a practical example of how Java can be used for real-world data management systems.