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)

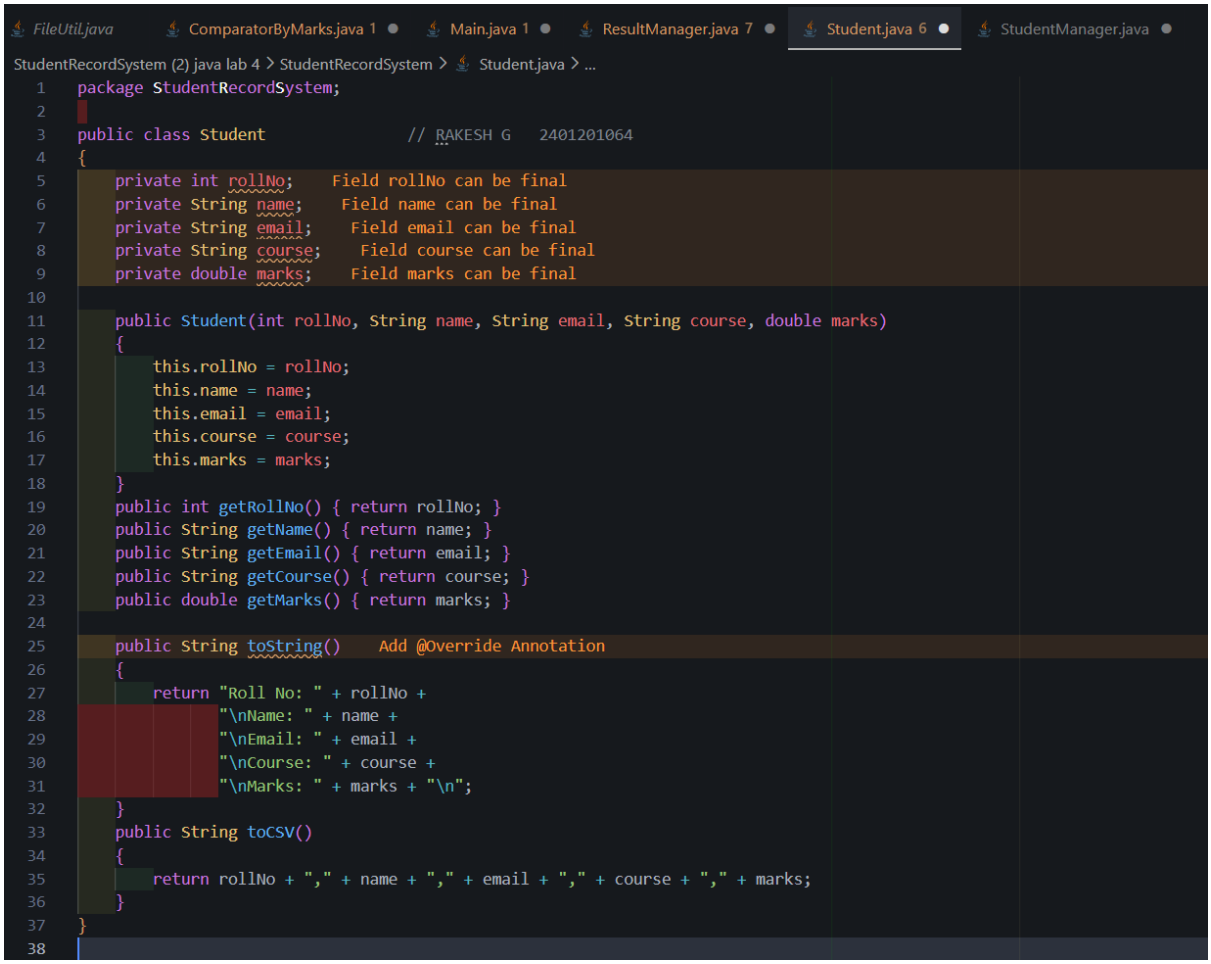
Submitted to:

Dr. Manish

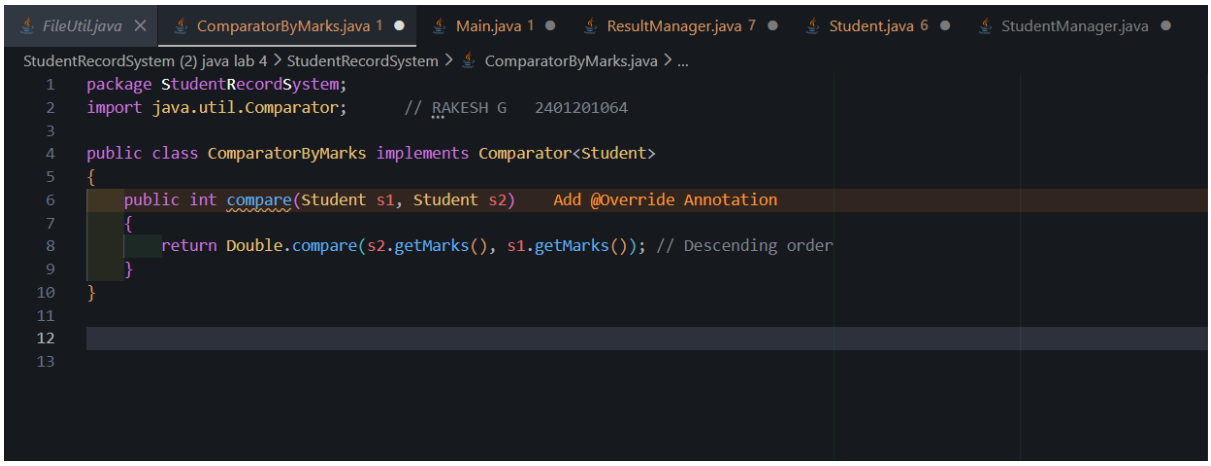
GitHub Repository:

https://github.com/rakesh4407/Java_Assignment_Rakesh

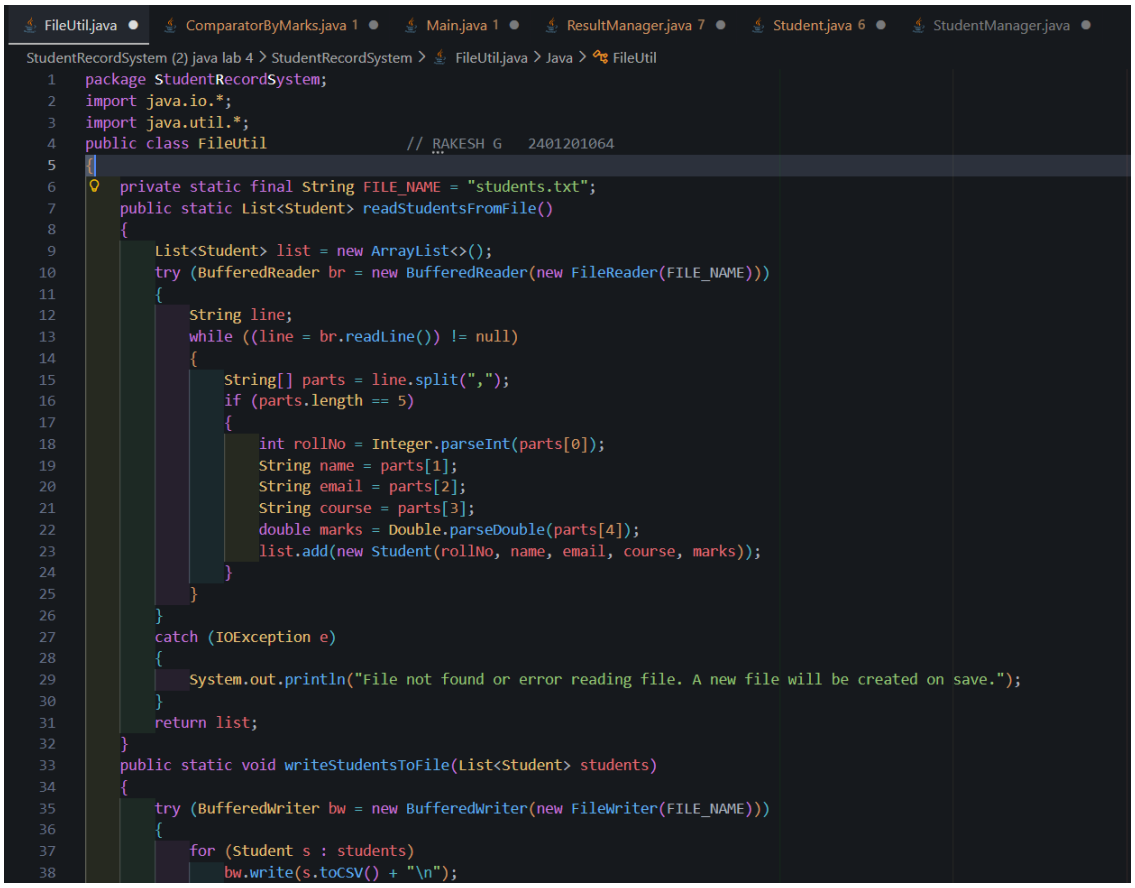
1. Student.java



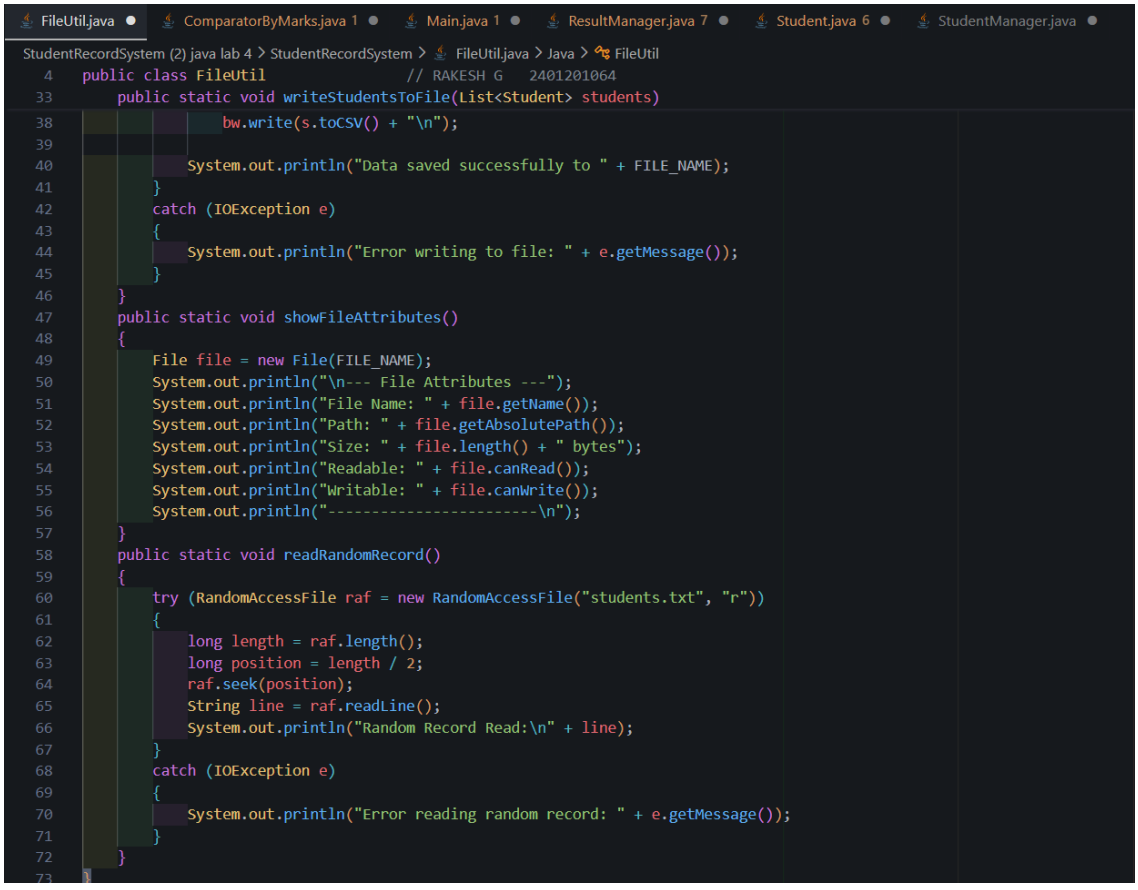
2. ComparatorByMarks.java



3. FileUtil.java

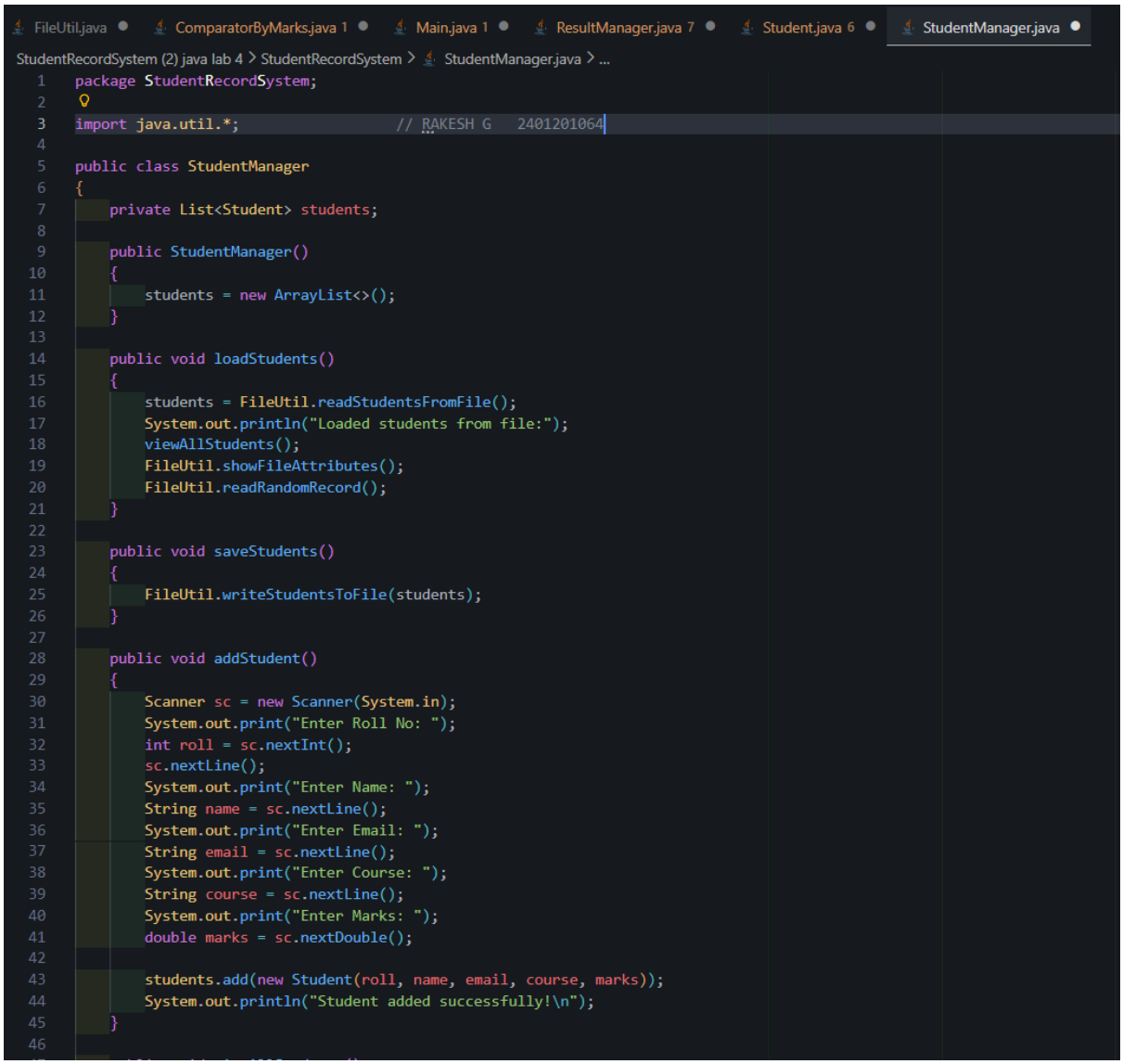


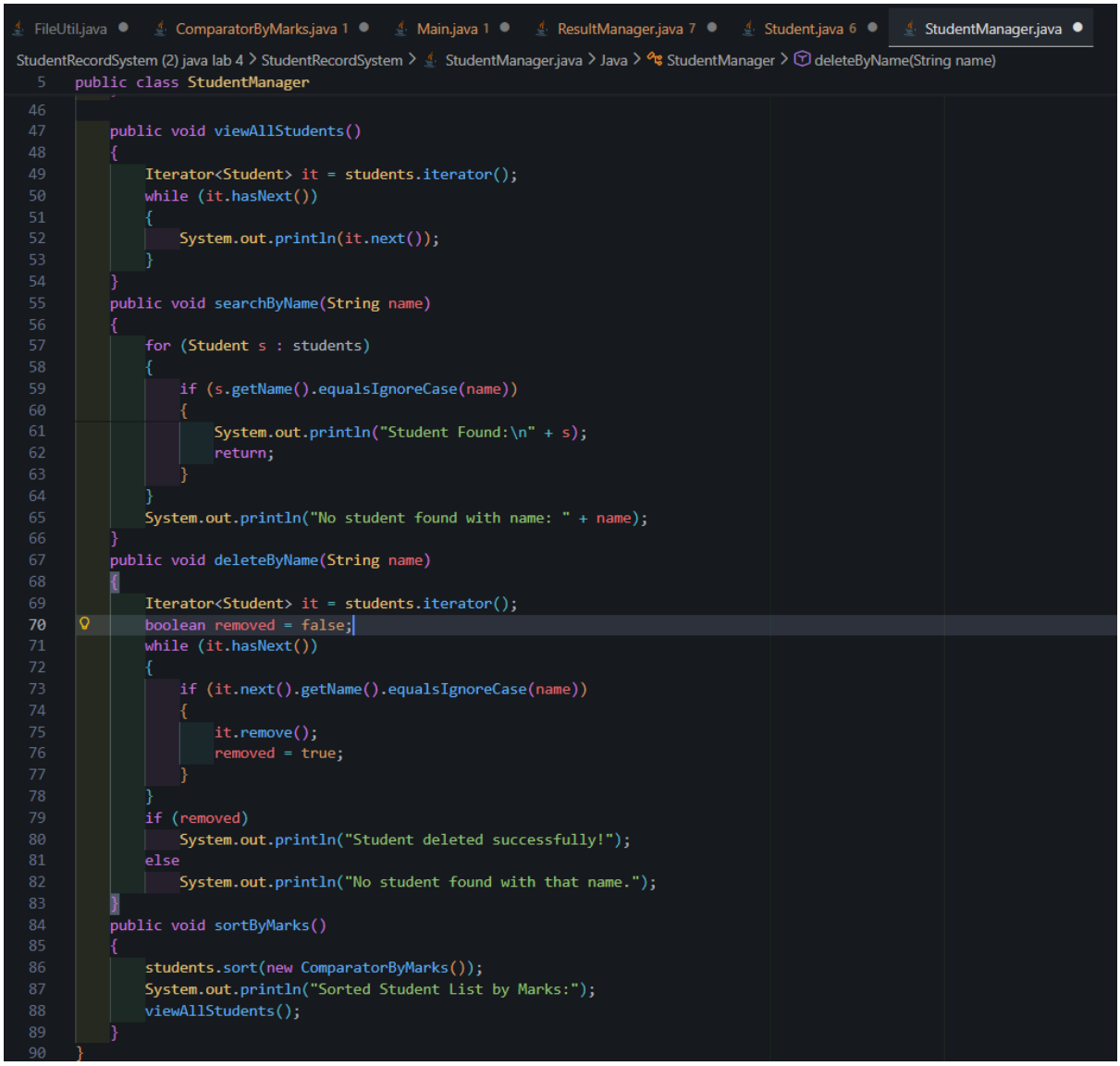
```java
package StudentRecordSystem;
import java.io.*;
import java.util.*;
public class FileUtil                // RAKESH G   2401201064
{
    private static final String FILE_NAME = "students.txt";
    public static List<Student> readStudentsFromFile()
    {
        List<Student> list = new ArrayList<>();
        try (BufferedReader br = new BufferedReader(new FileReader(FILE_NAME)))
        {
            String line;
            while ((line = br.readLine()) != null)
            {
                String[] parts = line.split(",");
                if (parts.length == 5)
                {
                    int rollNo = Integer.parseInt(parts[0]);
                    String name = parts[1];
                    String email = parts[2];
                    String course = parts[3];
                    double marks = Double.parseDouble(parts[4]);
                    list.add(new Student(rollNo, name, email, course, marks));
                }
            }
        }
        catch (IOException e)
        {
            System.out.println("File not found or error reading file. A new file will be created on save.");
        }
        return list;
    }
    public static void writeStudentsToFile(List<Student> students)
    {
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(FILE_NAME)))
        {
            for (Student s : students)
                bw.write(s.toCSV() + "\n");
```



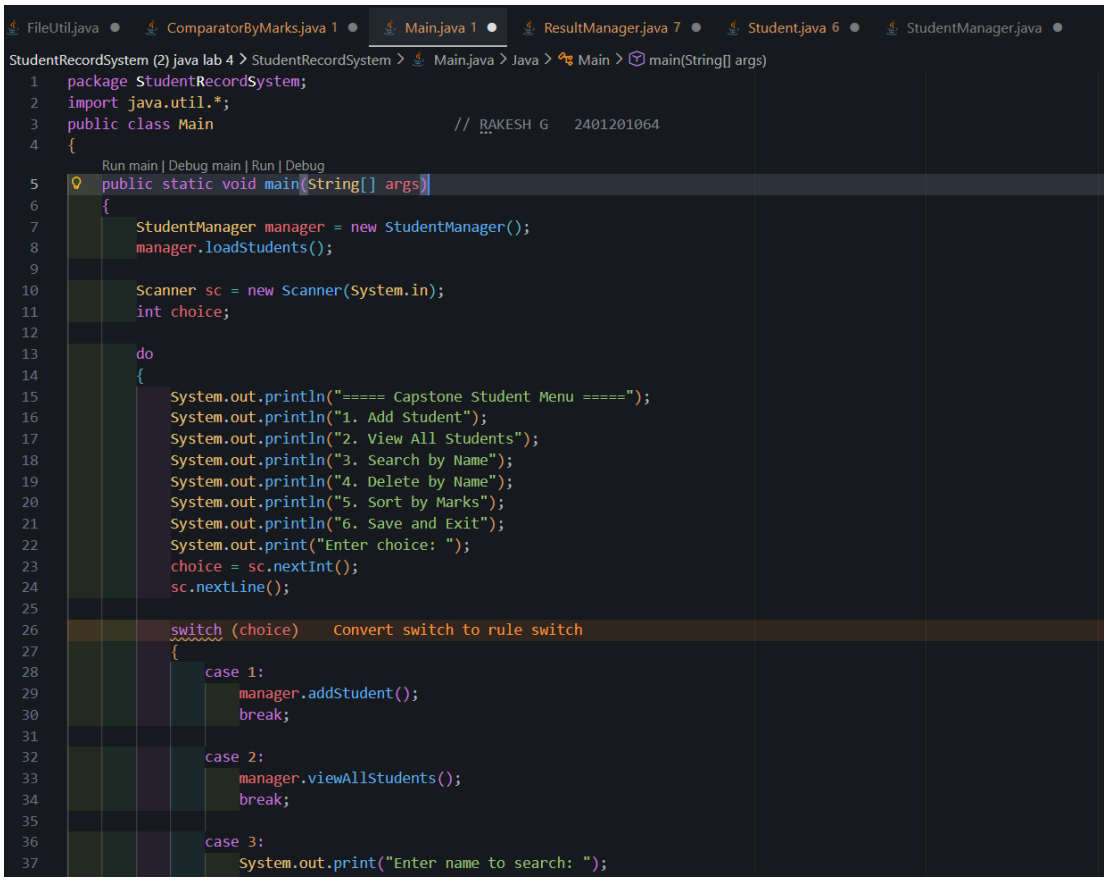
```java
public class FileUtil                // RAKESH G   2401201064
    public static void writeStudentsToFile(List<Student> students)
                bw.write(s.toCSV() + "\n");

            System.out.println("Data saved successfully to " + FILE_NAME);
        }
        catch (IOException e)
        {
            System.out.println("Error writing to file: " + e.getMessage());
        }
    }
    public static void showFileAttributes()
    {
        File file = new File(FILE_NAME);
        System.out.println("\n--- File Attributes ---");
        System.out.println("File Name: " + file.getName());
        System.out.println("Path: " + file.getAbsolutePath());
        System.out.println("Size: " + file.length() + " bytes");
        System.out.println("Readable: " + file.canRead());
        System.out.println("Writable: " + file.canWrite());
        System.out.println("-----------------------\n");
    }
    public static void readRandomRecord()
    {
        try (RandomAccessFile raf = new RandomAccessFile("students.txt", "r"))
        {
            long length = raf.length();
            long position = length / 2;
            raf.seek(position);
            String line = raf.readLine();
            System.out.println("Random Record Read:\n" + line);
        }
        catch (IOException e)
        {
            System.out.println("Error reading random record: " + e.getMessage());
        }
    }
}
```

4. StudentManager.java

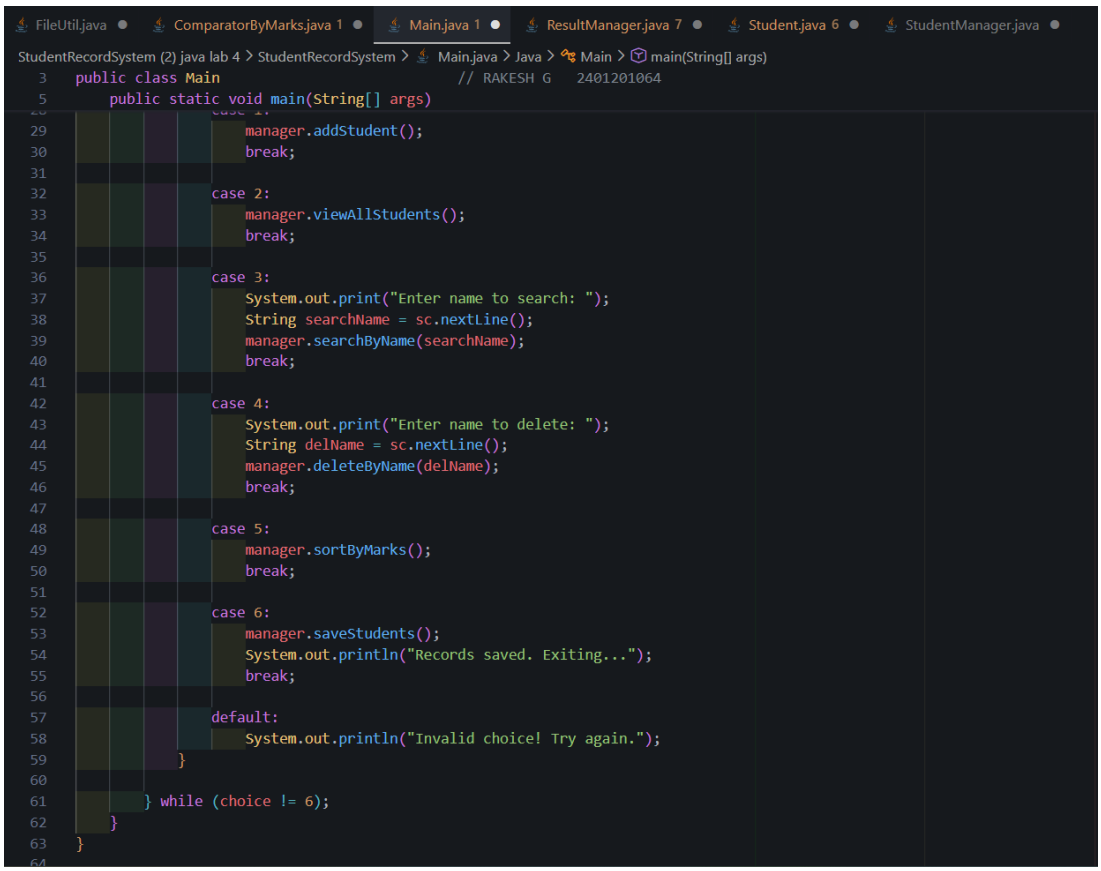



FileUtil.java
ComparatorByMarks.java 1
Main.java 1
ResultManager.java 7
Student.java 6
StudentManager.java
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > StudentManager.java > Java > StudentManager > deleteByName(String name)
5 public class StudentManager
46
47 public void viewAllStudents()
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

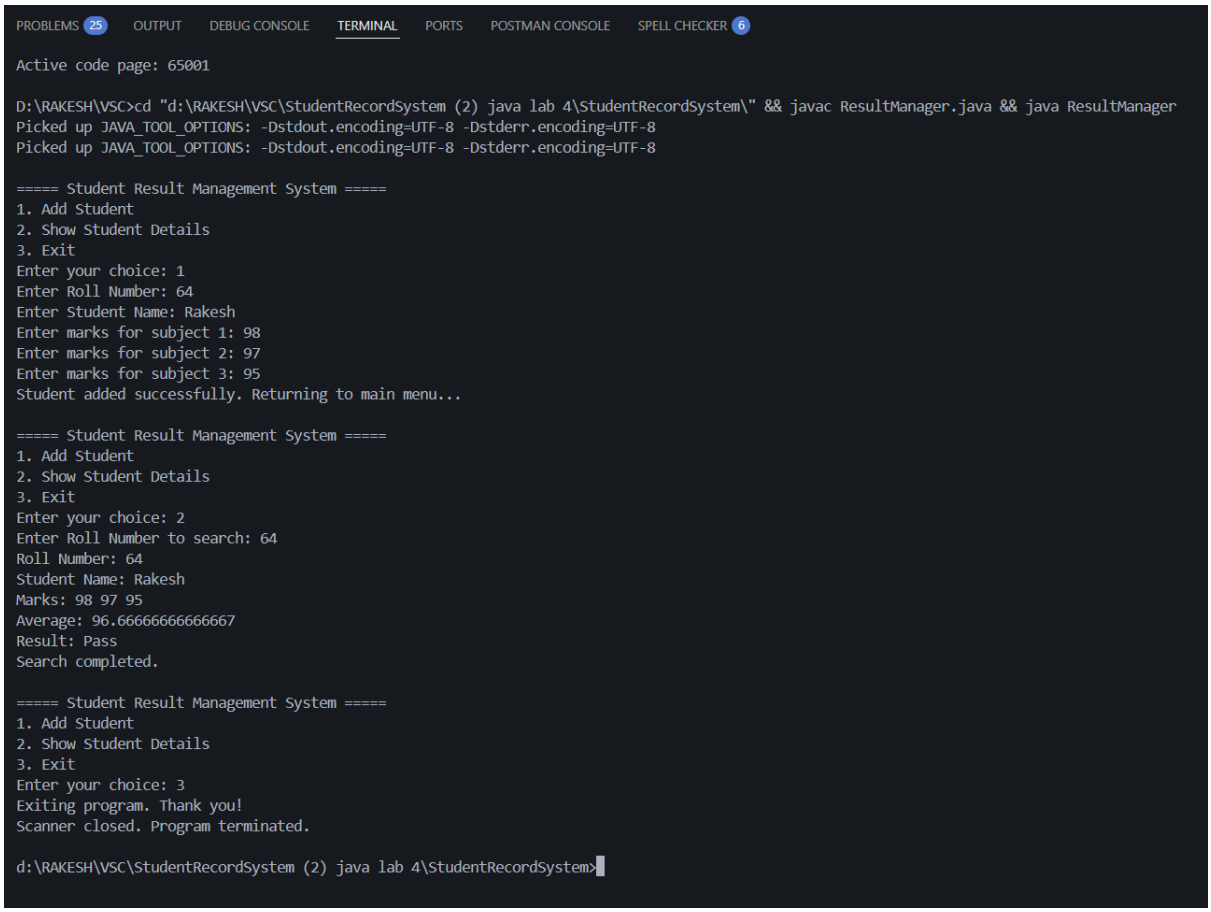


```
FileUtil.java  ComparatorByMarks.java 1  Main.java 1  ResultManager.java 7  Student.java 6  StudentManager.java
StudentRecordSystem (2) java lab 4 > StudentRecordSystem > Main.java > Java > Main > main(String[] args)
1   package StudentRecordSystem;
2   import java.util.*;
3   public class Main                          // RAKESH G   2401201064
4   {
        Run main | Debug main | Run | Debug
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
61          } while (choice != 6);
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.