K.R. Mangalam University
School of Engineering & Technology

Fundamentals Of Java Programming Lab

(ENCA203) Assignment 5

Student Management System

Submitted by:
Name: RAKESH G
Roll No: 2401201064

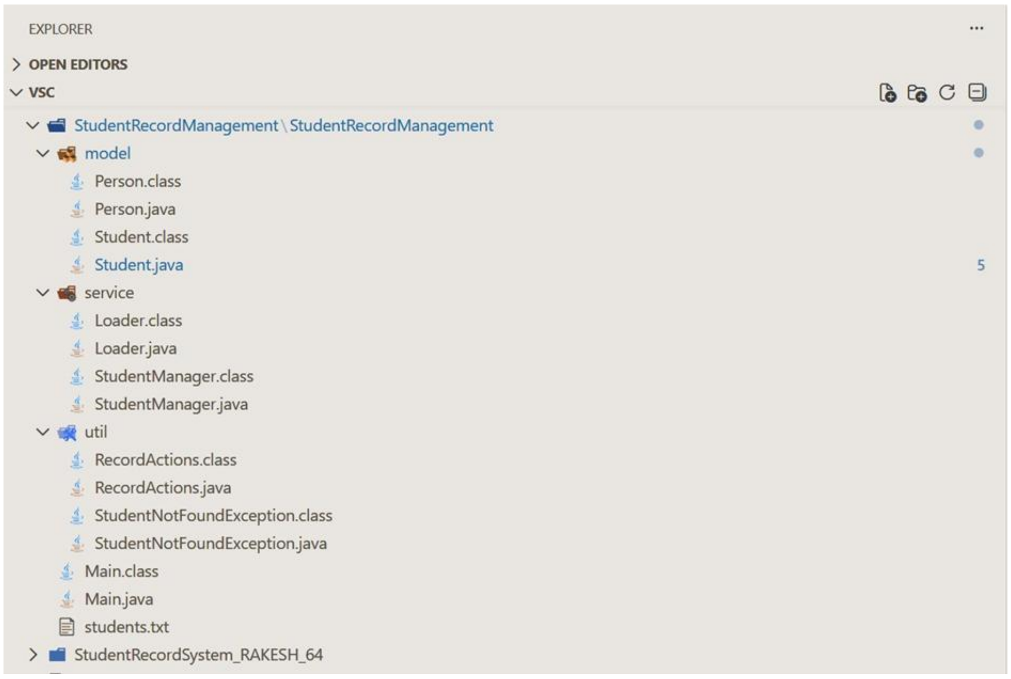
Class: BCA (AI & DS)

Submitted to:
Dr. Manish Kumar

GitHub Repository:

https://github.com/rakesh4407/Java_Assignment_Rakesh

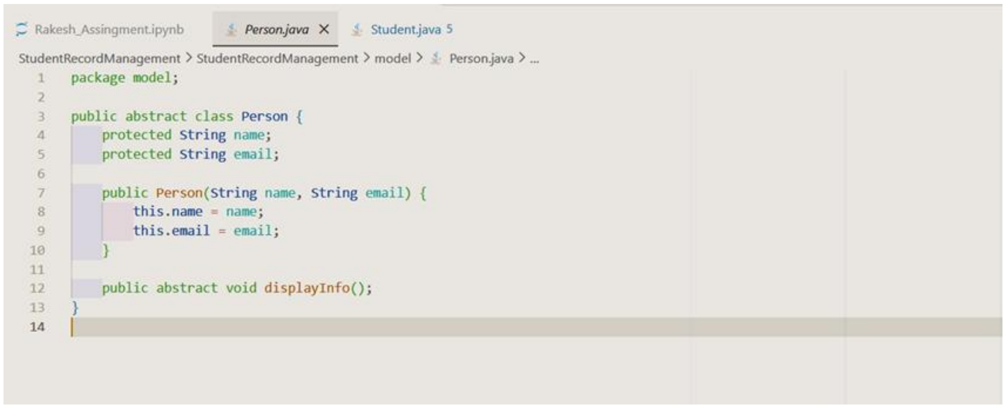
FOLDER STRUCTURE



```
EXPLORER
> OPEN EDITORS
v VSC
  v StudentRecordManagement\StudentRecordManagement
    v model
        Person.class
        Person.java
        Student.class
        Student.java                                  5
    v service
        Loader.class
        Loader.java
        StudentManager.class
        StudentManager.java
    v util
        RecordActions.class
        RecordActions.java
        StudentNotFoundException.class
        StudentNotFoundException.java
      Main.class
      Main.java
      students.txt
  > StudentRecordSystem_RAKESH_64
```

Code:

Person.java

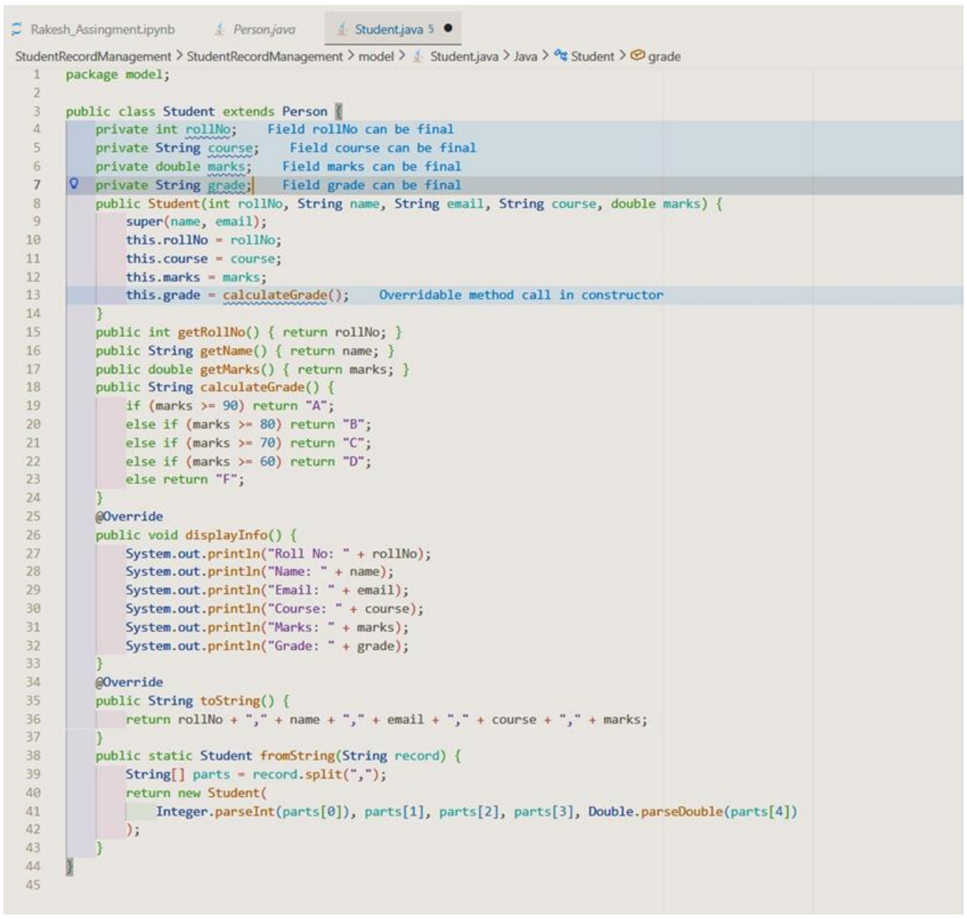


Rakesh_Assingment.ipynb   Person.java ×   Student.java 5

StudentRecordManagement > StudentRecordManagement > model > Person.java > ...

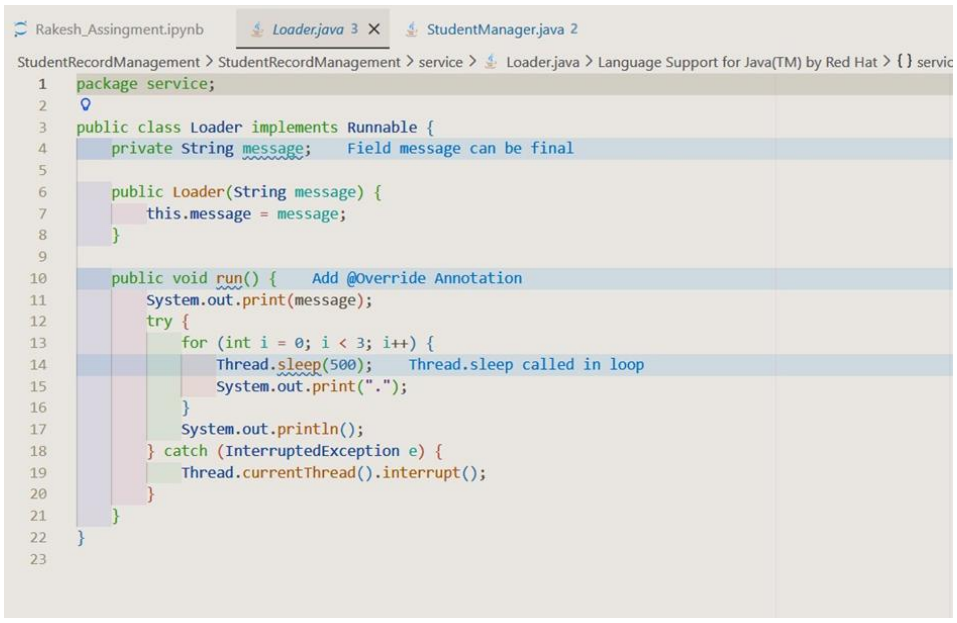
```java
package model;

public abstract class Person {
    protected String name;
    protected String email;

    public Person(String name, String email) {
        this.name = name;
        this.email = email;
    }

    public abstract void displayInfo();
}
```

Student.java


Rakesh_Assingment.ipynb
Person.java
Student.java 5 ●
StudentRecordManagement > StudentRecordManagement > model > Student.java > Java > Student > grade

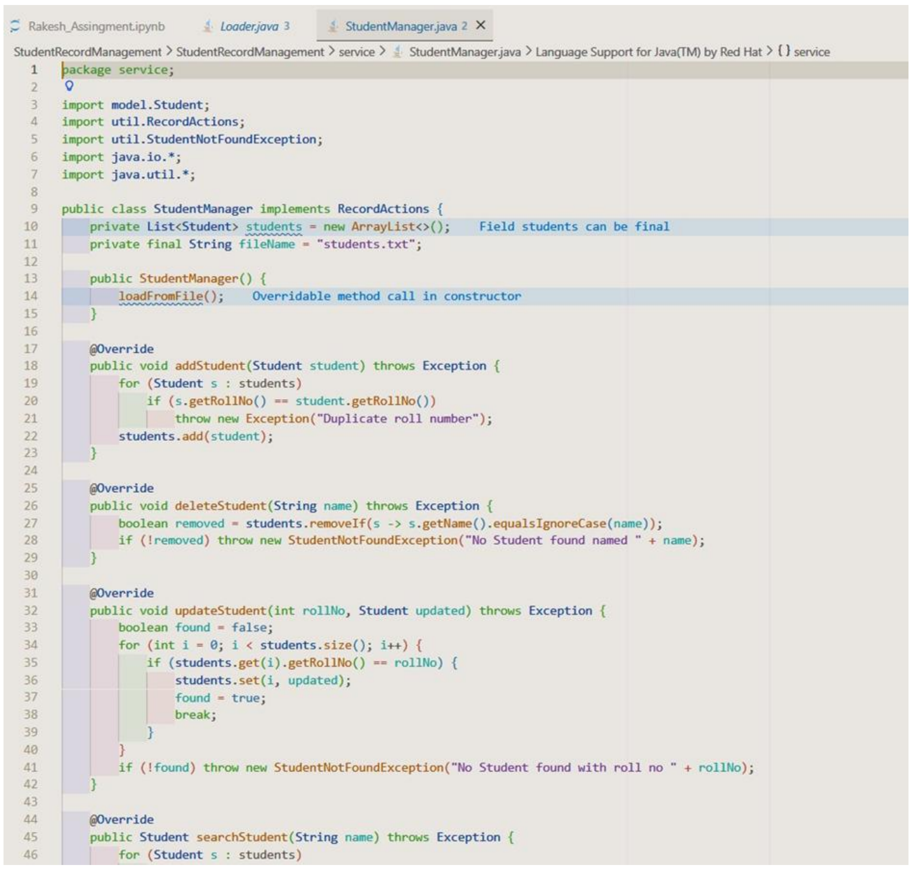
```java
package model;

public class Student extends Person {
    private int rollNo;    Field rollNo can be final
    private String course;    Field course can be final
    private double marks;    Field marks can be final
    private String grade;    Field grade can be final
    public Student(int rollNo, String name, String email, String course, double marks) {
        super(name, email);
        this.rollNo = rollNo;
        this.course = course;
        this.marks = marks;
        this.grade = calculateGrade();    Overridable method call in constructor
    }
    public int getRollNo() { return rollNo; }
    public String getName() { return name; }
    public double getMarks() { return marks; }
    public String calculateGrade() {
        if (marks >= 90) return "A";
        else if (marks >= 80) return "B";
        else if (marks >= 70) return "C";
        else if (marks >= 60) return "D";
        else return "F";
    }
    @Override
    public void displayInfo() {
        System.out.println("Roll No: " + rollNo);
        System.out.println("Name: " + name);
        System.out.println("Email: " + email);
        System.out.println("Course: " + course);
        System.out.println("Marks: " + marks);
        System.out.println("Grade: " + grade);
    }
    @Override
    public String toString() {
        return rollNo + "," + name + "," + email + "," + course + "," + marks;
    }
    public static Student fromString(String record) {
        String[] parts = record.split(",");
        return new Student(
            Integer.parseInt(parts[0]), parts[1], parts[2], parts[3], Double.parseDouble(parts[4])
        );
    }
}
```

Loader.java


Rakesh_Assingment.ipynb
Loader.java 3 ×
StudentManager.java 2
StudentRecordManagement > StudentRecordManagement > service > Loader.java > Language Support for Java(TM) by Red Hat > { } servic

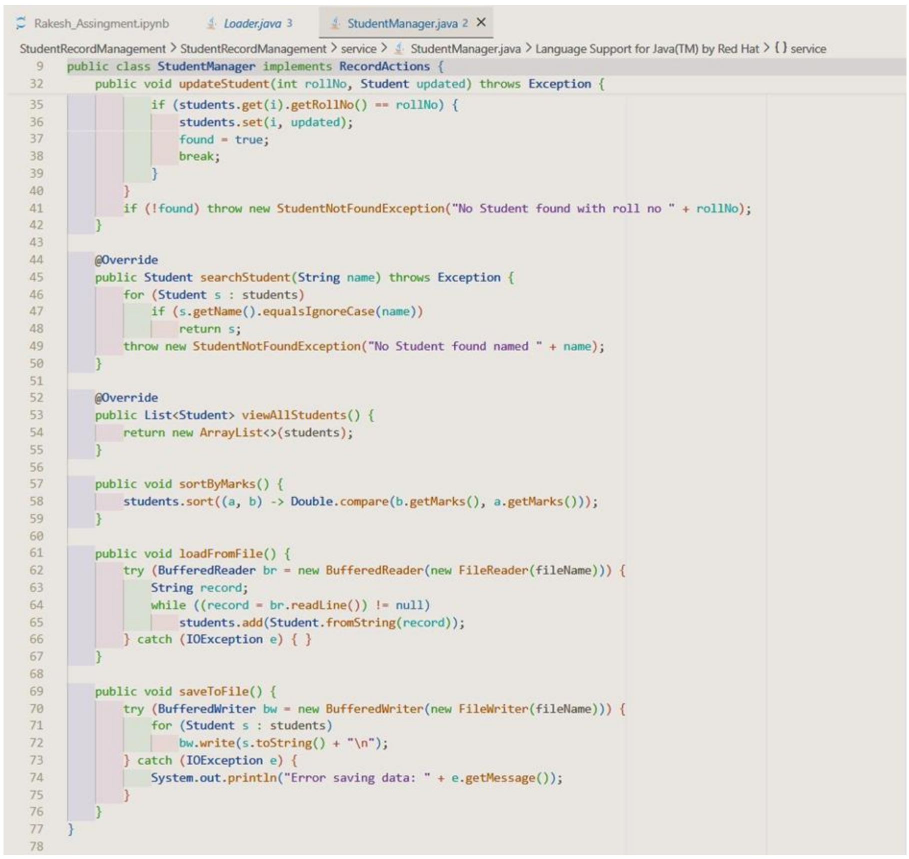
```java
package service;

public class Loader implements Runnable {
    private String message;    Field message can be final

    public Loader(String message) {
        this.message = message;
    }

    public void run() {    Add @Override Annotation
        System.out.print(message);
        try {
            for (int i = 0; i < 3; i++) {
                Thread.sleep(500);    Thread.sleep called in loop
                System.out.print(".");
            }
            System.out.println();
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
}
```

StudentManager.java



Rakesh_Assingment.ipynb
Loader.java 3
StudentManager.java 2 ×
StudentRecordManagement > StudentRecordManagement > service > StudentManager.java > Language Support for Java(TM) by Red Hat > { } service

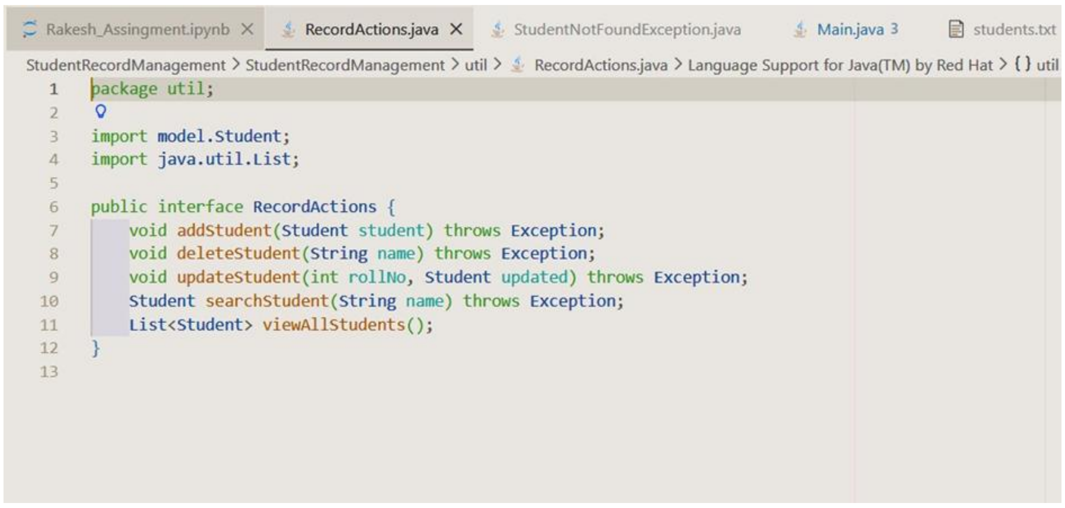
```java
package service;

import model.Student;
import util.RecordActions;
import util.StudentNotFoundException;
import java.io.*;
import java.util.*;

public class StudentManager implements RecordActions {
    private List<Student> students = new ArrayList<>();    Field students can be final
    private final String fileName = "students.txt";

    public StudentManager() {
        loadFromFile();    Overridable method call in constructor
    }

    @Override
    public void addStudent(Student student) throws Exception {
        for (Student s : students)
            if (s.getRollNo() == student.getRollNo())
                throw new Exception("Duplicate roll number");
        students.add(student);
    }

    @Override
    public void deleteStudent(String name) throws Exception {
        boolean removed = students.removeIf(s -> s.getName().equalsIgnoreCase(name));
        if (!removed) throw new StudentNotFoundException("No Student found named " + name);
    }

    @Override
    public void updateStudent(int rollNo, Student updated) throws Exception {
        boolean found = false;
        for (int i = 0; i < students.size(); i++) {
            if (students.get(i).getRollNo() == rollNo) {
                students.set(i, updated);
                found = true;
                break;
            }
        }
        if (!found) throw new StudentNotFoundException("No Student found with roll no " + rollNo);
    }

    @Override
    public Student searchStudent(String name) throws Exception {
        for (Student s : students)
```



Rakesh_Assingment.ipynb
Loader.java 3
StudentManager.java 2 ×
StudentRecordManagement > StudentRecordManagement > service > StudentManager.java > Language Support for Java(TM) by Red Hat > { } service

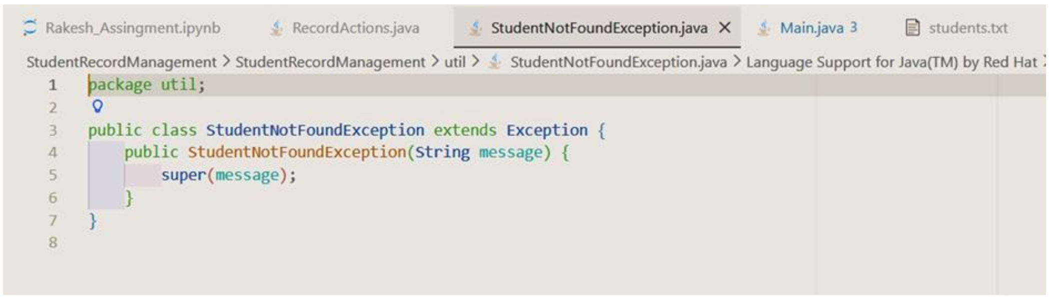
```java
public class StudentManager implements RecordActions {
    public void updateStudent(int rollNo, Student updated) throws Exception {
            if (students.get(i).getRollNo() == rollNo) {
                students.set(i, updated);
                found = true;
                break;
            }
        }
        if (!found) throw new StudentNotFoundException("No Student found with roll no " + rollNo);
    }

    @Override
    public Student searchStudent(String name) throws Exception {
        for (Student s : students)
            if (s.getName().equalsIgnoreCase(name))
                return s;
        throw new StudentNotFoundException("No Student found named " + name);
    }

    @Override
    public List<Student> viewAllStudents() {
        return new ArrayList<>(students);
    }

    public void sortByMarks() {
        students.sort((a, b) -> Double.compare(b.getMarks(), a.getMarks()));
    }

    public void loadFromFile() {
        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
            String record;
            while ((record = br.readLine()) != null)
                students.add(Student.fromString(record));
        } catch (IOException e) { }
    }

    public void saveToFile() {
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(fileName))) {
            for (Student s : students)
                bw.write(s.toString() + "\n");
        } catch (IOException e) {
            System.out.println("Error saving data: " + e.getMessage());
        }
    }
}
```

RecordActions.java


Rakesh_Assingment.ipynb
RecordActions.java
StudentNotFoundException.java
Main.java 3
students.txt
StudentRecordManagement > StudentRecordManagement > util > RecordActions.java > Language Support for Java(TM) by Red Hat > { } util

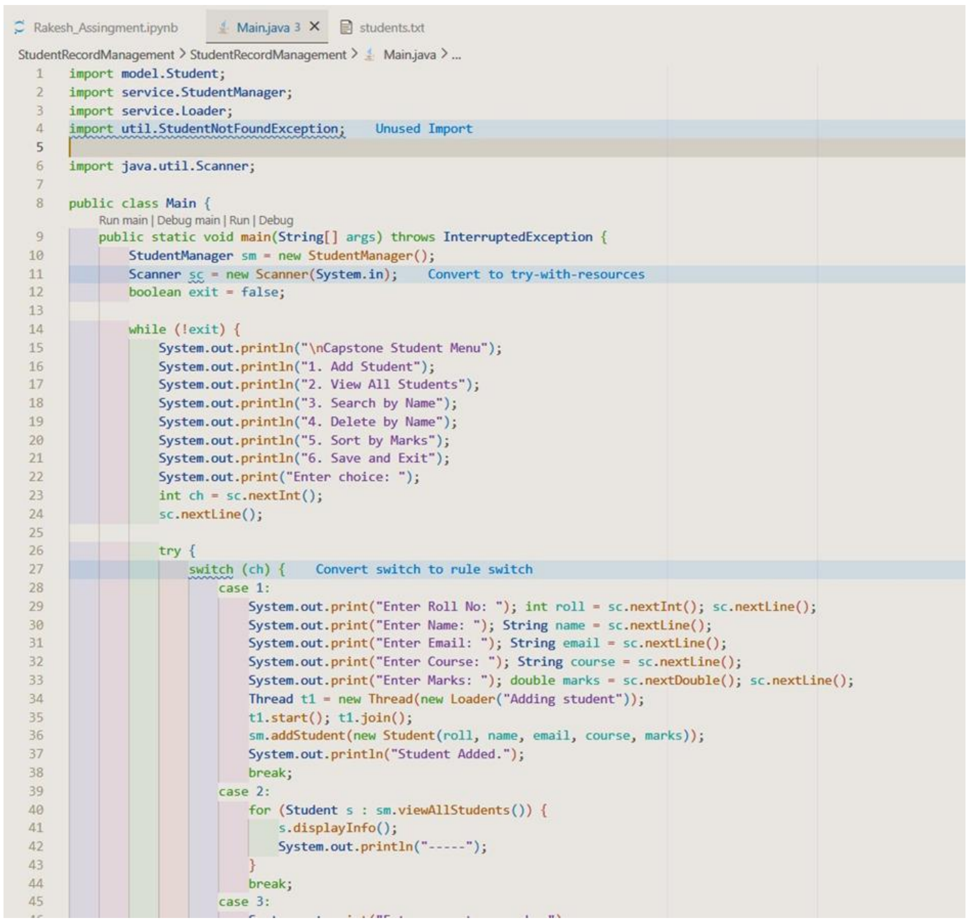
```java
package util;

import model.Student;
import java.util.List;

public interface RecordActions {
    void addStudent(Student student) throws Exception;
    void deleteStudent(String name) throws Exception;
    void updateStudent(int rollNo, Student updated) throws Exception;
    Student searchStudent(String name) throws Exception;
    List<Student> viewAllStudents();
}
```

StudentNotFoundException.java


Rakesh_Assingment.ipynb
RecordActions.java
StudentNotFoundException.java
Main.java 3
students.txt
StudentRecordManagement > StudentRecordManagement > util > StudentNotFoundException.java > Language Support for Java(TM) by Red Hat

```java
package util;

public class StudentNotFoundException extends Exception {
    public StudentNotFoundException(String message) {
        super(message);
    }
}
```

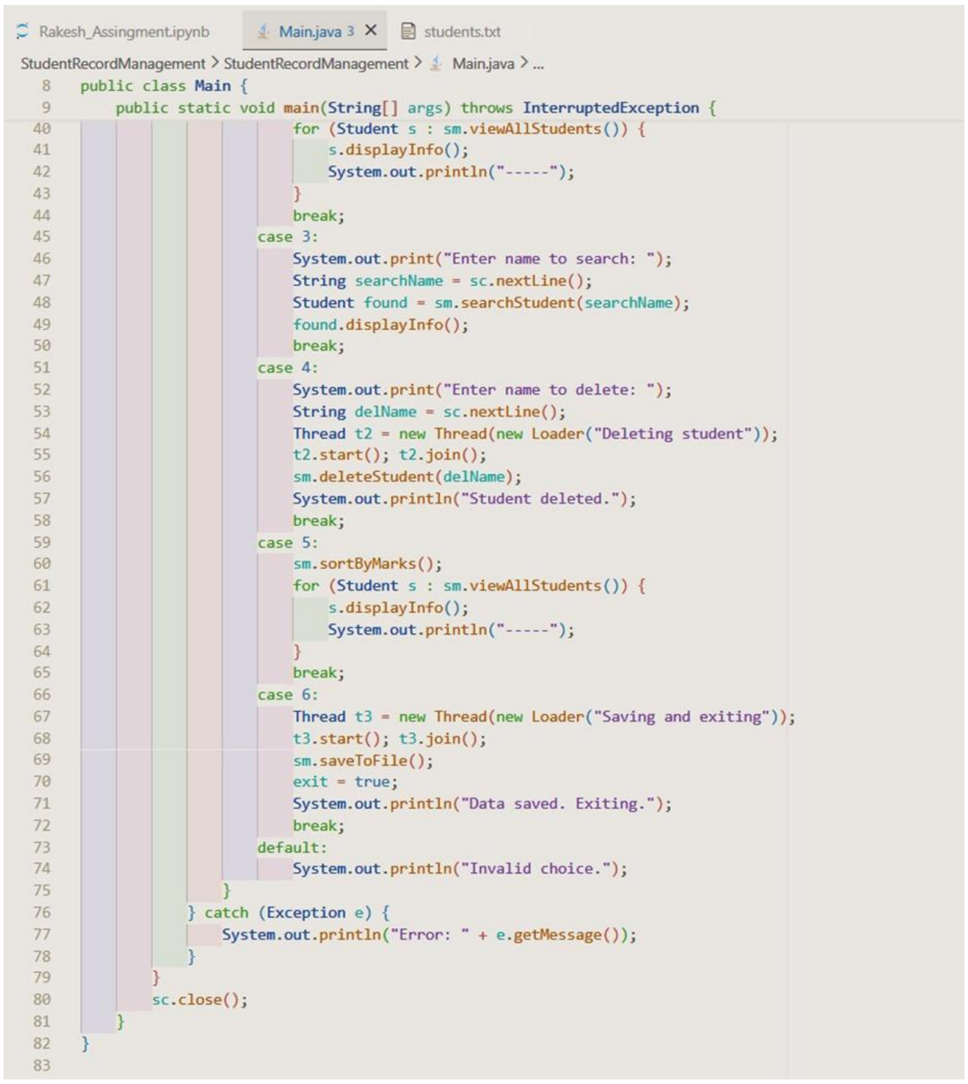
Main.java



Rakesh_Assingment.ipynb   Main.java 3 ×   students.txt

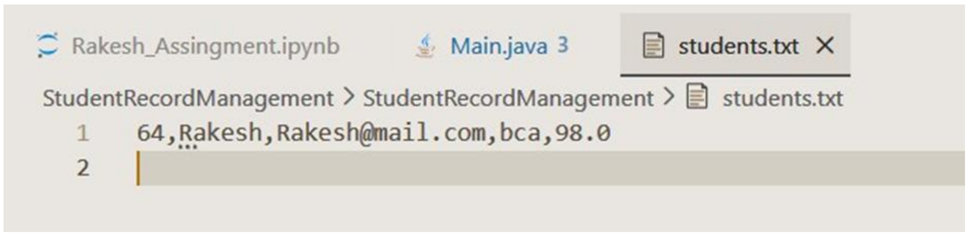
StudentRecordManagement > StudentRecordManagement > Main.java > ...

```java
import model.Student;
import service.StudentManager;
import service.Loader;
import util.StudentNotFoundException;    Unused Import

import java.util.Scanner;

public class Main {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws InterruptedException {
        StudentManager sm = new StudentManager();
        Scanner sc = new Scanner(System.in);    Convert to try-with-resources
        boolean exit = false;

        while (!exit) {
            System.out.println("\nCapstone Student Menu");
            System.out.println("1. Add Student");
            System.out.println("2. View All Students");
            System.out.println("3. Search by Name");
            System.out.println("4. Delete by Name");
            System.out.println("5. Sort by Marks");
            System.out.println("6. Save and Exit");
            System.out.print("Enter choice: ");
            int ch = sc.nextInt();
            sc.nextLine();

            try {
                switch (ch) {    Convert switch to rule switch
                    case 1:
                        System.out.print("Enter Roll No: "); int roll = sc.nextInt(); sc.nextLine();
                        System.out.print("Enter Name: "); String name = sc.nextLine();
                        System.out.print("Enter Email: "); String email = sc.nextLine();
                        System.out.print("Enter Course: "); String course = sc.nextLine();
                        System.out.print("Enter Marks: "); double marks = sc.nextDouble(); sc.nextLine();
                        Thread t1 = new Thread(new Loader("Adding student"));
                        t1.start(); t1.join();
                        sm.addStudent(new Student(roll, name, email, course, marks));
                        System.out.println("Student Added.");
                        break;
                    case 2:
                        for (Student s : sm.viewAllStudents()) {
                            s.displayInfo();
                            System.out.println("-----");
                        }
                        break;
                    case 3:
```


Rakesh_Assingment.ipynb
Main.java 3
students.txt
StudentRecordManagement > StudentRecordManagement > Main.java > ...

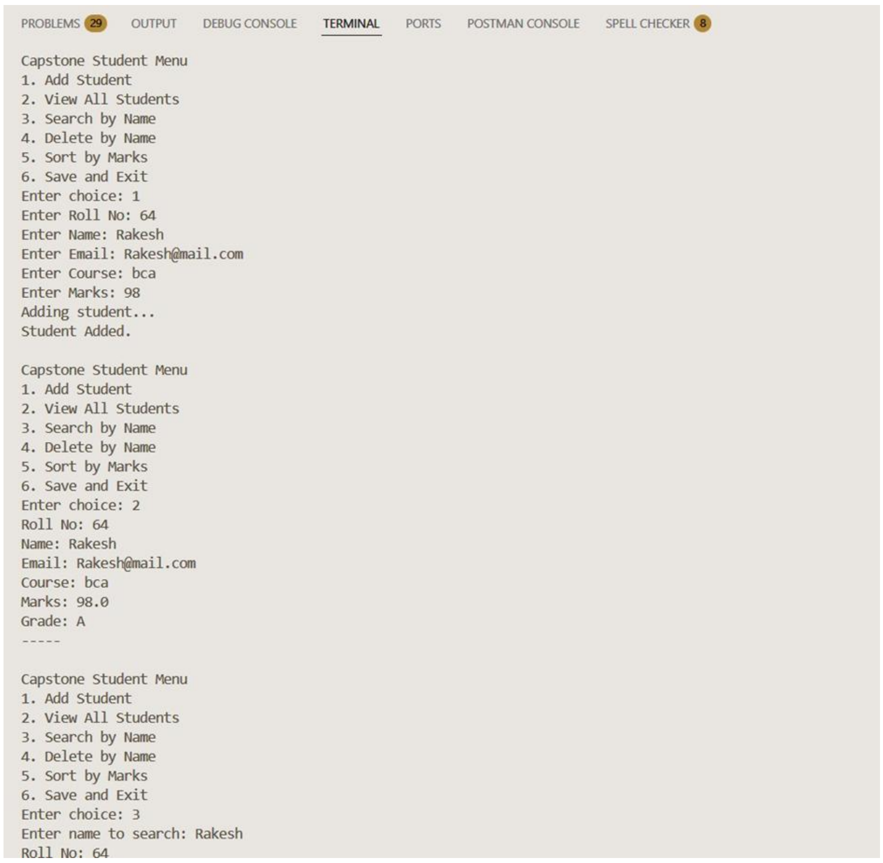
```java
public class Main {
    public static void main(String[] args) throws InterruptedException {
                        for (Student s : sm.viewAllStudents()) {
                            s.displayInfo();
                            System.out.println("-----");
                        }
                        break;
                    case 3:
                        System.out.print("Enter name to search: ");
                        String searchName = sc.nextLine();
                        Student found = sm.searchStudent(searchName);
                        found.displayInfo();
                        break;
                    case 4:
                        System.out.print("Enter name to delete: ");
                        String delName = sc.nextLine();
                        Thread t2 = new Thread(new Loader("Deleting student"));
                        t2.start(); t2.join();
                        sm.deleteStudent(delName);
                        System.out.println("Student deleted.");
                        break;
                    case 5:
                        sm.sortByMarks();
                        for (Student s : sm.viewAllStudents()) {
                            s.displayInfo();
                            System.out.println("-----");
                        }
                        break;
                    case 6:
                        Thread t3 = new Thread(new Loader("Saving and exiting"));
                        t3.start(); t3.join();
                        sm.saveToFile();
                        exit = true;
                        System.out.println("Data saved. Exiting.");
                        break;
                    default:
                        System.out.println("Invalid choice.");
                }
            } catch (Exception e) {
                System.out.println("Error: " + e.getMessage());
            }
        }
        sc.close();
    }
}
```

students.txt

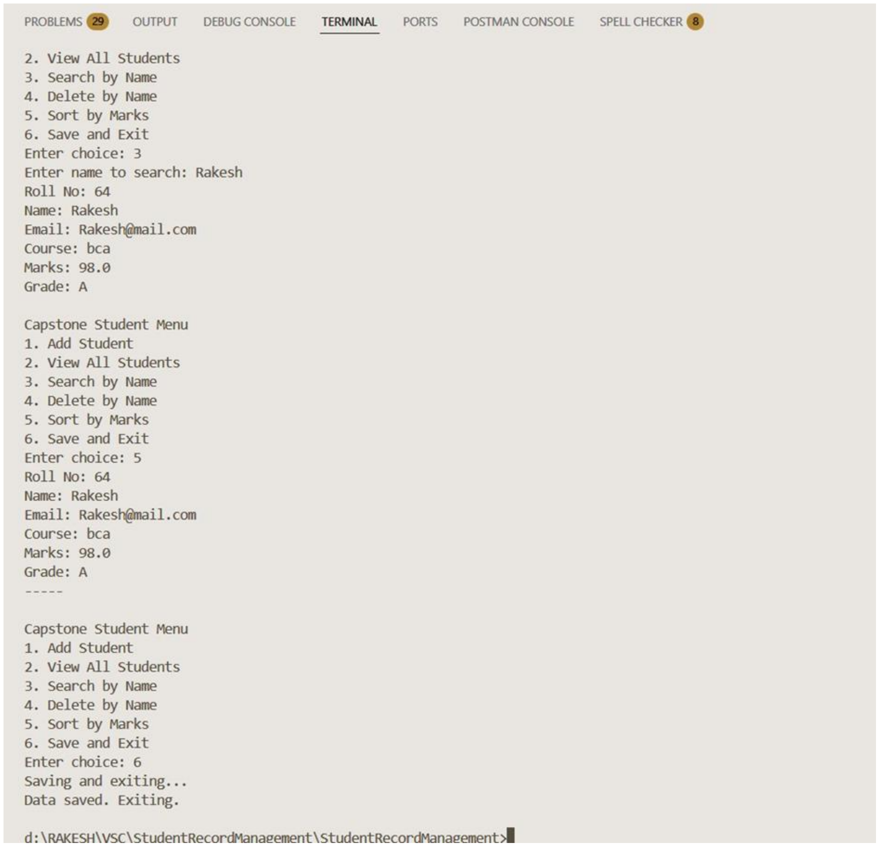

Rakesh_Assingment.ipynb
Main.java 3
students.txt
StudentRecordManagement > StudentRecordManagement > students.txt

```
64,Rakesh,Rakesh@mail.com,bca,98.0

```

Output:



```
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: 64
Enter Name: Rakesh
Enter Email: Rakesh@mail.com
Enter Course: bca
Enter Marks: 98
Adding student...
Student Added.

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: 2
Roll No: 64
Name: Rakesh
Email: Rakesh@mail.com
Course: bca
Marks: 98.0
Grade: A
-----

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: 3
Enter name to search: Rakesh
Roll No: 64
```



```
2. View All Students
3. Search by Name
4. Delete by Name
5. Sort by Marks
6. Save and Exit
Enter choice: 3
Enter name to search: Rakesh
Roll No: 64
Name: Rakesh
Email: Rakesh@mail.com
Course: bca
Marks: 98.0
Grade: A

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: 5
Roll No: 64
Name: Rakesh
Email: Rakesh@mail.com
Course: bca
Marks: 98.0
Grade: A
-----

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: 6
Saving and exiting...
Data saved. Exiting.

d:\RAKESH\VSC\StudentRecordManagement\StudentRecordManagement>
```

Explanation:

How to Compile and Run the Student Record Management System (One-Page Guide)

To run the Java-based Student Record Management System, you only need to compile the source code files (.java) and then execute the Main class, which contains the entry point of the program.

1. Navigate to the Project Folder

Open your terminal or command prompt and go to the folder where your project files are stored:

cd path/to/your/project

Example:

cd D:\RAKESH\VSC\StudentRecordManagement\StudentRecordManagement

2. Compile Java Source Files

Java only compiles .java files (source code), **not** .class files.

Run the following command:

javac *.java

This compiles all Java files in the directory and generates .class files automatically.

3. Run the Program

After successful compilation, run the program using the Main class: java

Main

This starts the Student Record Management System and displays the menu.

Why Main.java?

- Main.java contains the public static void main(String[] args) method.
- It is the **entry point** of the whole application.
- All other classes (Student, StudentManager, FileUtil, etc.) are helper classes and cannot run by themselves.