Assignment No. 08

Name -: Parth Manoj Poriya

Roll No-: 34

Branch-: SE IT(B2)

Subject-: Object Oriented Programming

TITLE

Write a program to Implement a program for maintaining a student records database
using File Handling. Student has Student_id, name, Roll_no, Class, marks and
address. Display the data for five students.
i) Create Database
ii)Display Database
iii) Clear Records
iv)Modify record
v)Search Record

INPUT

```
import java.io.*;
import java.util.*;
 class Student {
    private int studentId;
    private String name;
    private String rollNo;
```

```
    private String className;
    private double marks;
    private String address;

    public Student(int studentId, String name, String rollNo, String
className, double marks, String address) {
        this.studentId = studentId;
        this.name = name;
        this.rollNo = rollNo;
        this.className = className;
        this.marks = marks;
        this.address = address;
    }

    public String toString() {
        return studentId + "," + name + "," + rollNo + "," + className +
"," + marks + "," + address;
    }

    public static Student fromString(String line) {
        String[] parts = line.split(",");
        return new Student(
                Integer.parseInt(parts[0]),
                parts[1],
                parts[2],
                parts[3],
                Double.parseDouble(parts[4]),
                parts[5]
        );
    }

    public int getStudentId() {
        return studentId;
    }
}
```

```
public class Assign8{
   private static final String FILE_NAME = "students.txt";

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      while (true) {
         System.out.println("\n1. Add Student\n2. Display
Students\n3. Clear Records\n4. Modify Record\n5. Search
Record\n6. Exit");
         System.out.print("Choose an option: ");
         int choice = scanner.nextInt();
         scanner.nextLine(); // Consume newline

         switch (choice) {
            case 1: addStudent(scanner); break;
            case 2: displayStudents(); break;
            case 3: clearRecords(); break;
            case 4: modifyRecord(scanner); break;
            case 5: searchRecord(scanner); break;
            case 6: scanner.close(); return;
            default: System.out.println("Invalid choice.");
         }
      }
   }

   private static void addStudent(Scanner scanner) {
      try (BufferedWriter writer = new BufferedWriter(new
FileWriter(FILE_NAME, true))) {
         System.out.print("Enter Student ID: ");
         int id = scanner.nextInt();
         scanner.nextLine(); // Consume newline
         System.out.print("Enter Name: ");
         String name = scanner.nextLine();
```

```
            System.out.print("Enter Roll No: ");
            String rollNo = scanner.nextLine();
            System.out.print("Enter Class: ");
            String className = scanner.nextLine();
            System.out.print("Enter Marks: ");
            double marks = scanner.nextDouble();
            scanner.nextLine(); // Consume newline
            System.out.print("Enter Address: ");
            String address = scanner.nextLine();
            writer.write(new Student(id, name, rollNo, className, marks,
address).toString());
            writer.newLine();
            System.out.println("Student added.");
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    private static void displayStudents() {
        try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
            String line;
            System.out.println("Student Records:");
            while ((line = reader.readLine()) != null) {
                System.out.println(Student.fromString(line));
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    private static void clearRecords() {
        try {
            new PrintWriter(FILE_NAME).close();
            System.out.println("All records cleared.");
```

```
        } catch (FileNotFoundException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    private static void modifyRecord(Scanner scanner) {
        System.out.print("Enter Student ID to modify: ");
        int idToModify = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        List<Student> students = new ArrayList<>();
        boolean found = false;

        try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
            String line;
            while ((line = reader.readLine()) != null) {
                Student student = Student.fromString(line);
                if (student.getStudentId() == idToModify) {
                    found = true;
                    System.out.print("Enter new Name: ");
                    String name = scanner.nextLine();
                    System.out.print("Enter new Roll No: ");
                    String rollNo = scanner.nextLine();
                    System.out.print("Enter new Class: ");
                    String className = scanner.nextLine();
                    System.out.print("Enter new Marks: ");
                    double marks = scanner.nextDouble();
                    scanner.nextLine(); // Consume newline
                    System.out.print("Enter new Address: ");
                    String address = scanner.nextLine();
                    student = new Student(idToModify, name, rollNo,
classNameName, marks, address);
                }
                students.add(student);
            }
```

```
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }

        if (found) {
            try (BufferedWriter writer = new BufferedWriter(new
FileWriter(FILE_NAME))) {
                for (Student student : students) {
                    writer.write(student.toString());
                    writer.newLine();
                }
                System.out.println("Record modified.");
            } catch (IOException e) {
                System.out.println("Error: " + e.getMessage());
            }
        } else {
            System.out.println("Student ID not found.");
        }
    }

    private static void searchRecord(Scanner scanner) {
        System.out.print("Enter Student ID to search: ");
        int idToSearch = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        boolean found = false;

        try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
            String line;
            while ((line = reader.readLine()) != null) {
                Student student = Student.fromString(line);
                if (student.getStudentId() == idToSearch) {
                    System.out.println("Record Found: " + student);
                    found = true;
                    break;
```

```
            }
        }
    } catch (IOException e) {
        System.out.println("Error: " + e.getMessage());
    }

    if (!found) {
        System.out.println("Student ID not found.");
    }
  }
}
```

OUTPUT

```
PS C:\Users\Acer\OneDrive\Desktop\Java> javac Assign8.java
PS C:\Users\Acer\OneDrive\Desktop\Java> java Assign8

1. Add Student
2. Display Students
3. Clear Records
4. Modify Record
5. Search Record
6. Exit
Choose an option: 1
Enter Student ID: 121
Enter Name: Ankit
Enter Roll No: 1
Enter Class: 12th
Enter Marks: 80
Enter Address: Ankit@gmail.com
Student added.

1. Add Student
2. Display Students
```

3. Clear Records
4. Modify Record
5. Search Record
6. Exit
Choose an option: 2
Student Records:
121,Ankit,1,12th,80.0,Ankit@gmail.com

1. Add Student
2. Display Students
3. Clear Records
4. Modify Record
5. Search Record
6. Exit
Choose an option: 4
Enter Student ID to modify: 121
Enter new Name: Vijay
Enter new Roll No: 2
Enter new Class: 12th
Enter new Marks: 75
Enter new Address: Vijay@gmail.com
Record modified.

1. Add Student
2. Display Students
3. Clear Records
4. Modify Record
5. Search Record
6. Exit
Choose an option: 3
All records cleared.
