K.R. Mangalam University

School of Engineering & Technology

Lab Assignment 2

Java Programming

Submitted by: Submitted to:

Name: Rakesh G   Dr. Manish Kumar

Roll No: 2401201064

Class: BCA (AI & DS)

GitHub Repository:

https://github.com/rakesh4407/Java_Assignment_Rakesh

CODE:

```java
import java.util.*;

// Abstract Class Person
abstract class Person {

    protected String name;
    protected String email;

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

    // Abstract Method
    public abstract void displayInfo();
}

// Student Class (extends Person)
class Student extends Person {

    private int rollNo;
    private String course;
    private double marks;        Variable marks is never read
    private String grade;        Variable grade is never read
    private String researchArea;  // optional for M.Tech students

    // Constructor 1 (without research area)
    public Student(int rollNo, String name, String email, String course, double marks, String grade) {
        super(name, email);
        this.rollNo = rollNo;
        this.course = course;
        this.marks = marks;
        this.grade = grade;
    }

    // Constructor 2 (with research area) → Overloading
    public Student(int rollNo, String name, String email, String course, double marks, String grade, String researchArea) {
```

```java
20  class Student extends Person {
41      }
42
43      // Method Overriding
44      @Override
45      public void displayInfo() {
46          System.out.println(x:"Student Info:");
47          System.out.println("Roll No: " + rollNo);
48          System.out.println("Name: " + name);
49          System.out.println("Email: " + email);
50          System.out.println("Course: " + course);
51          if (researchArea != null) {
52              System.out.println("Research Area: " + researchArea);
53          }
54          System.out.println();
55      }
56
57      // Method Overloading (polymorphism)
58      public void displayInfo(boolean simple) {
59          if (simple) {
60              System.out.println(x:"Student Info:");
61              System.out.println("Roll No: " + rollNo);
62              System.out.println("Name: " + name);
63              System.out.println("Email: " + email);
64              System.out.println("Course: " + course);
65              System.out.println();
66          } else {
67              displayInfo();
68          }
69      }
70
71      // Final method
72      public final void showFinalMessage() {
73          System.out.println(x:"This is a final method in a final class.");
74      }
75
76      // finalize method
77      @Override
```

```java
20  class Student extends Person {
78      protected void finalize() throws Throwable {    finalize() declared
79          System.out.println(x:"Finalize method called before object is garbage collected.");
80      }
81
82      // Getters & Setters
83      public int getRollNo() {
84          return rollNo;
85      }
86
87      public void setCourse(String course) {
88          this.course = course;
89      }
90
91      public void setMarks(double marks) {
92          this.marks = marks;
93      }
94
95      public void setGrade(String grade) {
96          this.grade = grade;
97      }
98  }
99
100 // Interface RecordActions
101 interface RecordActions {
102
103     void addStudent(Student student);
104
105     void deleteStudent(int rollNo);
106
107     void updateStudent(int rollNo, String course, double marks, String grade);
108
109     Student searchStudent(int rollNo);
110
111     List<Student> viewAllStudents();
112 }
113
114 // StudentManager (implements RecordActions)
```

```java
115    class StudentManager implements RecordActions {
118
119        public StudentManager() {
120            studentMap = new HashMap<>();
121        }
122
123        @Override
124        public void addStudent(Student student) {
125            if (studentMap.containsKey(student.getRollNo())) {
126                System.out.println("Student with Roll No " + student.getRollNo() + " already exists!");
127            } else {
128                studentMap.put(student.getRollNo(), student);
129                System.out.println(x:"Student added successfully.");
130            }
131        }
132
133        @Override
134        public void deleteStudent(int rollNo) {
135            if (studentMap.remove(rollNo) != null) {
136                System.out.println(x:"Student removed successfully.");
137            } else {
138                System.out.println(x:"Student not found.");
139            }
140        }
141
142        @Override
143        public void updateStudent(int rollNo, String course, double marks, String grade) {
144            Student student = studentMap.get(rollNo);
145            if (student != null) {
146                student.setCourse(course);
147                student.setMarks(marks);
148                student.setGrade(grade);
149                System.out.println(x:"Student updated successfully.");
150            } else {
151                System.out.println(x:"Student not found.");
152            }
153        }
154
```

```java
115    class StudentManager implements RecordActions {
155        @Override
156        public Student searchStudent(int rollNo) {
157            return studentMap.get(rollNo);
158        }
159
160        @Override
161        public List<Student> viewAllStudents() {
162            return new ArrayList<>(studentMap.values());
163        }
164    }
165
166    // Main Class
167    public class Inheritance {
168
           Run main | Debug main | Run | Debug
169        public static void main(String[] args) {
170            StudentManager manager = new StudentManager();
171
172            // Adding Students
173            Student s1 = new Student(rollNo:101, name:"Rakesh", email:"Rakesh@mail.com", course:"BCA", marks:95, grade:"A+");
174            Student s2 = new Student(rollNo:102, name:"Riya", email:"riya@mail.com", course:"M.Tech", marks:80, grade:"A", resear…"AI");
175
176            manager.addStudent(s1);
177            manager.addStudent(s2);
178
179            // Viewing Students
180            for (Student s : manager.viewAllStudents()) {
181                s.displayInfo();
182            }
183
184            // Using Overloaded method
185            System.out.println(x:"[Note] Overloaded display method:");
186            s1.displayInfo(simple:true);
187
188            // Final Method
189            s1.showFinalMessage();
190
```

```java
167    public class Inheritance {
168        public static void main(String[] args) {
181                    s.displayInfo();
182                }
183
184            // Using Overloaded method
185            System.out.println(x:"[Note] Overloaded display method:");
186            s1.displayInfo(simple:true);
187
188            // Final Method
189            s1.showFinalMessage();
190
191            // Searching
192            Student searchResult = manager.searchStudent(rollNo:102);
193            if (searchResult != null) {
194                System.out.println(x:"\nFound Student:");
195                searchResult.displayInfo();
196            }
197
198            // Updating
199            manager.updateStudent(rollNo:101, course:"B.Tech CSE", marks:88, grade:"A+");
200
201            // Deleting
202            manager.deleteStudent(rollNo:102);
203
204            // Final list
205            System.out.println(x:"\nFinal Student List:");
206            for (Student s : manager.viewAllStudents()) {
207                s.displayInfo();
208            }
209
210            // Garbage Collection to call finalize()
211            s1 = null;      The assigned value is never used
212            s2 = null;      The assigned value is never used
213            System.gc(); // Suggest JVM to run garbage collector
214        }
215    }
216
```

OUTPUT:

```
Student added successfully.
Student added successfully.
Student Info:
Roll No: 101
Name: Rakesh
Email: Rakesh@mail.com
Course: BCA

Student Info:
Roll No: 102
Name: Riya
Email: riya@mail.com
Course: M.Tech
Research Area: AI

[Note] Overloaded display method:
Student Info:
Roll No: 101
Name: Rakesh
Email: Rakesh@mail.com
Course: BCA

This is a final method in a final class.

Found Student:
Student Info:
Roll No: 102
Name: Riya
Email: riya@mail.com
Course: M.Tech
Research Area: AI

Student updated successfully.
Student removed successfully.

Final Student List:
Student Info:
Roll No: 101
Name: Rakesh
Email: Rakesh@mail.com
Course: B.Tech CSE


d:\RAKESH\VSC>
```