

JAVA ASSIGNMENT 2 (LAB)

NAME-BHAVYA RATTAN

ROLL NO-2401201004

COURSE-BCA(AI& DS)

CODE:-

```
import model.Student;
```

```
import service.StudentManager;
```

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        StudentManager manager = new StudentManager();
```

```
        Scanner sc = new Scanner(System.in);
```

```
        while (true) {
```

```
            System.out.println("\n--- Student Management System ---");
```

```
            System.out.println("1. Add Student");
```

```
            System.out.println("2. Delete Student");
```

```
            System.out.println("3. Update Student");
```

```
            System.out.println("4. Search Student");
```

```
            System.out.println("5. View All Students");
```

```
            System.out.println("6. Exit");
```

```
            System.out.print("Enter choice: ");
```

```
            int choice = sc.nextInt();
```

```
            sc.nextLine(); // consume newline
```

```
switch (choice) {  
    case 1:  
        System.out.print("Roll No: ");  
        int rollNo = sc.nextInt();  
        sc.nextLine();  
        System.out.print("Name: ");  
        String name = sc.nextLine();  
        System.out.print("Email: ");  
        String email = sc.nextLine();  
        System.out.print("Course: ");  
        String course = sc.nextLine();  
        System.out.print("Marks: ");  
        double marks = sc.nextDouble();  
        Student student = new Student(rollNo, name, email, course, marks);  
        manager.addStudent(student);  
        break;  
    case 2:  
        System.out.print("Enter Roll No to delete: ");  
        manager.deleteStudent(sc.nextInt());  
        break;  
    case 3:  
        System.out.print("Enter Roll No to update: ");  
        int rNo = sc.nextInt();  
        sc.nextLine();  
        System.out.print("New Course: ");  
        String newCourse = sc.nextLine();  
        System.out.print("New Marks: ");
```

```

        double newMarks = sc.nextDouble();

        manager.updateStudent(rNo, newCourse, newMarks);

        break;
    case 4:

        System.out.print("Enter Roll No to search: ");

        Student s = manager.searchStudent(sc.nextInt());

        if (s != null) s.displayInfo();

        else System.out.println("Student not found.");

        break;
    case 5:

        manager.viewAllStudents();

        break;
    case 6:

        System.out.println("Exiting...");

        sc.close();

        System.exit(0);

    default:

        System.out.println("Invalid choice!");

    }

}

}

}

```

MODEL:-

(STUDENT.JAVA)

package model;

```

public class Student extends Person {

```

```
private int rollNo;

private String course;

private double marks;

private char grade;


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();

}
```

// Method Overloading: displayInfo with optional research area

```
public void displayInfo() {

    System.out.println("Student Info:");

    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);

    System.out.println("-----");

}
```

```
public void displayInfo(String researchArea) {

    displayInfo();

    System.out.println("Research Area: " + researchArea);

}
```

```

        System.out.println("-----");
    }

    private char calculateGrade() {
        if (marks >= 90) return 'A';
        else if (marks >= 75) return 'B';
        else if (marks >= 60) return 'C';
        else if (marks >= 50) return 'D';
        else return 'F';
    }

    public int getRollNo() {
        return rollNo;
    }

    public void setCourse(String course) {
        this.course = course;
    }

    public void setMarks(double marks) {
        this.marks = marks;
        this.grade = calculateGrade();
    }
}

```

(PERSON.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;  
    }  
  
    // Abstract method to be implemented by subclasses  
    public abstract void displayInfo();  
}
```

SERVICE:-

(RECORDACTIONS.JAVA)

```
package service;
```

```
import model.Student;
```

```
public interface RecordActions {  
    void addStudent(Student student);  
    void deleteStudent(int rollNo);  
    void updateStudent(int rollNo, String course, double marks);  
    Student searchStudent(int rollNo);  
    void viewAllStudents();  
}
```

(STUDENTMANAGER.JAVA)

```
package service;
```

```
import model.Student;
```

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
public class StudentManager implements RecordActions {  
    private Map<Integer, Student> students = new HashMap<>();
```

```
    // Add student if rollNo is unique
```

```
    public void addStudent(Student student) {  
        if (students.containsKey(student.getRollNo())) {  
            System.out.println("Error: Duplicate roll number!");  
            return;  
        }  
        students.put(student.getRollNo(), student);  
        System.out.println("Student added successfully.");  
    }
```

```
    public void deleteStudent(int rollNo) {  
        if (students.remove(rollNo) != null) {  
            System.out.println("Student deleted successfully.");  
        } else {  
            System.out.println("Student not found.");  
        }  
    }  
}
```

```
public void updateStudent(int rollNo, String course, double marks) {  
    Student s = students.get(rollNo);  
    if (s != null) {  
        s.setCourse(course);  
        s.setMarks(marks);  
        System.out.println("Student updated successfully.");  
    } else {  
        System.out.println("Student not found.");  
    }  
}  
  
public Student searchStudent(int rollNo) {  
    return students.get(rollNo);  
}  
  
public void viewAllStudents() {  
    if (students.isEmpty()) {  
        System.out.println("No students found.");  
        return;  
    }  
    for (Student s : students.values()) {  
        s.displayInfo();  
    }  
}
```

OUTPUT


```
PS C:\Users\Dell\OneDrive\ドキュメント\StudentManagementSystem\
C:\Users\Dell\OneDrive\ドキュメント\StudentManagementSystem
```

```
>> java -cp ".\bin" Main
```

```
>>
```

```
--- Student Management System ---
```

1. Add Student
2. Delete Student
3. Update Student
4. Search Student
5. View All Students
6. Exit

```
Enter choice: 1
```

```
Roll No: 2
```

```
Name: bhavya
```

```
Email: bhavya
```

```
Course: bca
```

```
Marks: 70
```

```
Student added successfully.
```

```
--- Student Management System ---
```