Solution for Lab 11.
Solution 1 with three classes:

Class Student:

```
public class Student {
	private int id;
	private String name;
	private double score;

	public Student () {}
	public Student (int id, String name, double score) {
		this.id=id;
		this.name=name;
		this.score=score;
	}
	public void setId(int id) {
		this.id = id;
	}
	public void setName(String name) {
		this.name = name;
	}
	public void setScore(double score) {
		this.score = score;
	}
	public int getId() {
		return id;
	}
	public String getName() {
		return name;
	}
	public double getScore() {
		return score;
	}
}
```

Class courseManager:

```
public class courseManager {
	private Student [] students;
	private int nStudents;
	public static final int MAX_SIZE=100;
	public courseManager () {
		students = new Student [MAX_SIZE];
	}
	public int getNStudents() {
		return nStudents;
	}
	public void addStudent(Student newStudent) {
		if (nStudents < MAX_SIZE) {
			students[nStudents] = newStudent;
			nStudents++;
		} else
			System.out.println("ERROR: COURSE IS FULL");
	}
	public void addStudent2(Student newStudent) {
		if (nStudents<MAX_SIZE) {
			if (findStudentById(newStudent.getId())==-1) {
				students[nStudents]=newStudent;
				nStudents++;
			} else
				System.out.println("Student is already added");
		} else
			System.out.println("ERROR: COURSE IS FULL");
	}
	public void dispalyStudent(int index) {
		System.out.println("["+index+"]: ID="+students[index].getId()+
				", Name="+students[index].getName()+",
Score="+students[index].getScore());
	}
	public int findStudentByName(String name) {
		for (int i = 0 ; i<nStudents;i++)
			if (students[i].getName().equalsIgnoreCase(name))
				return i;
		return -1;
	}
	public int findStudentById(int id) {
		for (int i = 0 ; i<nStudents;i++)
			if (students[i].getId() == id )
				return i;
		return -1;
	}
	public int findMaxScoreIndex() {
		int index=0;
		double max=students[index].getScore();
		for (int i=1; i<nStudents;i++)
```

```
                if (students[i].getScore()>max) {
                      max=students[i].getScore();
                      index=i;
                }
          return index;
    }
    public double computeAverageScore() {
          double sum=0;
          for (int i=0; i<nStudents;i++) sum = sum + students[i].getScore();
          return sum/nStudents;
    }
    public void removeStudent(int index) {
          students[index]=students[nStudents-1];
          students[nStudents-1]=null;
          nStudents--;
    }
    public void removeAndShiftStudents(int index) {
          for (int i = index;i<nStudents;i++)
                students[i]=students[i+1];

          students[nStudents-1]=null;
          nStudents--;
    }
}
```

Class testCourseManager

```
import java.util.Scanner;
public class testCourseManager {
	public static void main(String[] args) {
		Scanner input = new Scanner (System.in);
		courseManager cm = new courseManager();
		System.out.print("Enter the number of Students :");
		int N = input.nextInt();
		for (int i=0;i<N;i++) {
			System.out.print("Please enter the ID, name, and score of student
"+i+" : " );
			Student s = new
Student(input.nextInt(),input.next(),input.nextDouble());
			cm.addStudent(s);
		}
		System.out.println("Students are: ");
		for (int i=0;i<cm.getNStudents();i++)
			cm.dispalyStudent(i);

		System.out.println("The Average score for all students :"
			+ cm.computeAverageScore());

		System.out.println("The student with Max Score is ");
		cm.dispalyStudent(cm.findMaxScoreIndex());

		//	Remove student
		System.out.print("Enter index of student you want to delete: ");
		int d = input.nextInt();
		cm.removeStudent(d);
		System.out.println("Student "+d+" was deleted");
		System.out.println("After deletion, The students are: ");
		for (int i=0;i<cm.getNStudents();i++)
			cm.dispalyStudent(i);
	}
}
```