



Software Engineering Lab

Home ► My courses ► Previous Years ► 2020-21 ► Spring Semester (2020-21) ►
Software Engineering Lab ► Class Test 1 ► Programming Re-Test

Started on Wednesday, 3 February 2021, 2:15 PM

State Finished

Completed on Wednesday, 3 February 2021, 2:57 PM

Time taken 42 mins 31 secs

Grade 50.00 out of 50.00 (100%)

Question 1

Complete

Mark 50.00 out of
50.00

Flag question

Consider a situation where you have to design a software for college admission where candidates from different school boards can be ranked for admission according to their percentile scores. The percentile score of a candidate is the percentage of students who have scored less marks than the candidate in the same exam.

Each board has a marks distribution, stored in terms of marks of each percentile of student. For example, Central board may have 50th and 60th percentile marks of 65 and 70, whereas State board may have 50th and 60th percentile marks of 60 and 67.

Given the marks of a candidate, his / her percentile score can be calculated by linearly interpolating between the two bounding percentile marks of the corresponding board. For example, if a Central board candidate's mark is 67, his / her percentile score is 54.

Design and write a Java object-oriented program that allows the user to input:

- A list of boards along with their percentile marks
- A list of candidates with their board name and marks

And outputs a ranked list of candidates according to their percentile scores. You have to design a console-based interface that allows the user to input and view this information.

Marks distribution:

- Program design (Classes, objects, functions, etc) - 20%
- User interface design and implementation (intuitiveness and ease of use) - 10%
- Implementation of features (the tasks to be performed) - 70%

```

import java.util.*;

public class retest {
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    HashMap<String, Board> boards=new HashMap<>();
    Vector<Student> students=new Vector<>();
    int op=0;
    while(op!=4){
        System.out.print("\nSELECT OPTION FROM LIST BELOW:\n\n" +
        "1> Add a board.\n" +
        "2> Add a student.\n" +
        "3> Get sorted list.\n" +
        "4> Exit\n\n" +
        "OPTION NUMBER: "); op = input.nextInt();
        input.nextLine();
        String name;
        switch(op) {
case 1:
            System.out.println("Enter name of board:");
            name=input.nextLine();
            if(boards.containsKey(name))
                System.out.println("Board already exists.\n");
            else
                boards.put(name, new Board(input,name));
            break;
case 2:
            System.out.println("Enter name of board:");
            name=input.nextLine();
            if(boards.containsKey(name))
                students.add(new Student(input, name, boards));
            else
                System.out.println("Board doesn't exist.\n");
            break;
case 3:
            Collections.sort(students);
            for(Student s:students){
                System.out.println(s.name+" "+s.percentage);
            }
            break;
case 4:
            System.out.println("\n\tBye!");
            break;
default:
            System.out.println("\n\tEnter a valid option number.");
        }
    }
}

```

```

    }
}

class Board{
    String name;
    int lower_percentile, lower_marks, upper_percentile, upper_marks;
    public Board(Scanner input, String board){
        this.name=board;
        System.out.println("Enter lower bound percentile: ");
        lower_percentile=input.nextInt();
        System.out.println("Enter lower bound marks: ");
        lower_marks=input.nextInt();
        System.out.println("Enter upper bound percentile: ");
        upper_percentile=input.nextInt();
        System.out.println("Enter upper bound marks: ");
        upper_marks=input.nextInt();
    }
    int getpercentile(int marks){
        double slope=(double)(upper_percentile-lower_percentile)/(double)(upper_marks-lower_marks);
        return (int)(upper_percentile+(slope*(marks-upper_marks)));
    }
}

class Student implements Comparable<Student>{
    String name, board;
    int marks, percentile;
    public Student(Scanner input, String board, HashMap<String, Board> boards){
        this.board=board;
        System.out.println("Enter name of student: ");
        name=input.nextLine();
        System.out.println("Enter marks of student: ");
        marks=input.nextInt();
        percentile=boards.get(board).getpercentile(marks);
    }

    @Override
    public int compareTo(Student student) {
        if(percentile> student.percentile)
            return 1;
        else
            return -1;
    }
}

```

Comment:

Finish review

QUIZ NAVIGATION

1

Finish review

You are logged in as Nisarg Upadhyaya (Log out)
CS29006_S2021