Noel LeJeune e-mail: lejeunen@mscd.edu SI 1052

Program04: Interactive Grade Book

Performance Measures

1. Demonstrate that you can use iteration
2. Demonstrate that you can use ArrayList to work with a collection of objects

Scenario

Miss Farnsworth, your favorite spinster 3rd grade teacher asks you to create a grade book program that allows her to record student scores for each student in her class. She wants to enter the grades for a bunch of students into the program. When she finishes entering all the students' information, the program displays every student's information. A single student's information is displayed on two lines.

Requirements Details

- She identifies each student by name
- The number of scores for each student is not known in advance (she enters scores until she decides there are no more scores to enter for that student)
- The grades for a student are between 0 and 100 inclusive
- The number of students is not known in advance – she enters each student's information, one at a time, until she is done (i.e., no more students to process). Enter quit as the student name to indicate no more students.
- When she is done, the program displays a list of every student's information
- Student information output has the following format. The first line has the student's name (only the first name – no spaces allowed in the name) followed by each score that was entered for that student (each score is separated by a space). The second line also has the student's name followed by the number of scores, the highest score, the lowest score, and the total of the scores.

Technical Requirements

- Good object-oriented design and programming requires at least two classes (a Student class and a GradeBook class)
- **You must use an ArrayList in the GradeBook class for Student objects**
- **You must use an ArrayList in the Student class for the scores (data type is Double)**
- Modularization (i.e., Methods) must be used in an appropriate object-oriented approach
    - ***Write methods (in the Student class) for finding the lowest score, the highest score, and the total of all the scores – do not calculate as the data is entered.***

Submit - Use the "Assignments Drop Box" in Blackboard - do NOT submit via email

Create a ***zip file*** containing all the required files *Name the file with your name, e.g.* **`SueJones04.zip`**

Submit ALL files in one zip file consisting of:
1. Your properly documented **source** code– ***do not*** submit the byte code files (i.e. .class files)
2. A sample of the output from running your program (`Output04.txt`)
3. The answers to the "Standard Questions" (`StandardQuestions04.txt`)

Standard Questions

A. How much time did you spend on this assignment?

B. What difficulties did you experience?

C. What were the results of running your test cases/program? Run the program enough times to see that it works correctly – more about testing in subsequent assignments.

Example Program Execution

```
$ java GradeBook

Enter next student's name (or enter quit to end): Amelia
Next score (-1 to end): 99.9
Next score (-1 to end): 88.8
Next score (-1 to end): 77.7
Next score (-1 to end): -1
Enter next student's name (or enter quit to end): Barbie
Next score (-1 to end): 55.5
Next score (-1 to end): 66.6
Next score (-1 to end): 77.7
Next score (-1 to end): 88.8
Next score (-1 to end): -1
Enter next student's name (or enter quit to end): Chloe
Next score (-1 to end): 100.0
Next score (-1 to end): 0
Next score (-1 to end): -1
Enter next student's name (or enter quit to end): quit

Grade Book Report
--------------------------
Amelia 99.9 88.8 77.7
Amelia 3 99.9 77.7 266.4
Barbie 55.5 66.6 77.7 88.8
Barbie 4 88.8 55.5 288.6
Chloe 100.0 0
Chloe 2 100.0 0.0 100.0
```