Programming Assignment 7/8

Scenario:

You are working for the U.S. Olympic Committee on a program to help eliminate judge corruption during event scoring for events that require subjective judging. Currently, all judge scores count. The USOC would like to create a program so that the lowest and highest score is dropped. For example, if scores earned are: 1.6, 6.7, 8.2, 9.8, and 6.3, the 1.6 and 9.8 would be dropped and the total score would be 21.2.

Create a ***modular*** program that will allow five judges to each enter a score between 0.0 and 10.0, inclusive. When a judge enters an invalid score, he/she must be informed of an error and re-prompted. After all judges have entered their scores, the total score should be calculated with the lowest and highest score dropped. Based on the total score earned, the athlete's Olympic readiness is determined as follows:

- A score below 24.0 indicates the athlete is not ready for the Olympics
- A score at 24.0, up to 27.0 not inclusive, indicates the athlete could earn a bronze medal
- A score at 27.0, up to 28.5 not inclusive, indicates the athlete could earn a silver medal
- A score at 28.5, up to 30.0, inclusive, indicates the athlete could earn a gold medal

Upon determining the athlete's ranking, print out a well-formatted report that contains all five scores earned, the total score earned with lowest and highest scores dropped, and the athlete's Olympic readiness.

Other Requirements:

- You must check that all numerical input is valid.
- Your solution must use one or more arrays/ArrayLists.

To Do (Check Blackboard for Due Dates):

Programming Assignment 7: Solution Design

1) Create a defining diagram that shows the input, processing, and output
2) Create a structure diagram/hierarchy chart grouping processes from the defining diagram into ***modules*** (You need to do this, but you do not need to turn it in as part of the grade)
3) Create a ***modular*** solution algorithm using pseudocode
4) Show testing using the desk checking table method, to include test data, expected results, and a desk checking table. Make sure your desk checking considers multiple cases including both valid and invalid test data to prove your algorithm will work

Upload a Word document containing only items above to Blackboard.

Grading Criteria	
Requirement	**Points**
Defining Diagram with input, processing, and output	40
Efficient Solution Algorithm	40
Thorough Desk Checking Table including test data, and expected results	20

Full points will be awarded for an accurate, efficient, complete defining diagram, solution algorithm, and desk checking table. Partial credit is available.

Programming Assignment 8: Solution Implementation

Write a well-documented, efficient Java program that implements the algorithm you identified. Include appropriate documentation as identified in the documentation expectations document.

Note: You may not use the Scanner or System.out classes. For input/output, you must use the JOptionPane class.

Upload the .java file of the final program to Blackboard.

Full points will be awarded for an accurate, efficient, complete Java program. Partial credit is available. <u>Any final program that does not compile will receive an automatic zero.</u>

Grading Criteria	
Requirement	**Points**
Implementation of Java Program, using efficient practices where appropriate, such as the use of constants, good variable names, no redundant code, etc.	70
Appropriate objective-style documentation	10
Appropriate intermediate comments	20