Lab Assignment #8

Due date: 12:00 pm, 4/3/2017

Problem Description

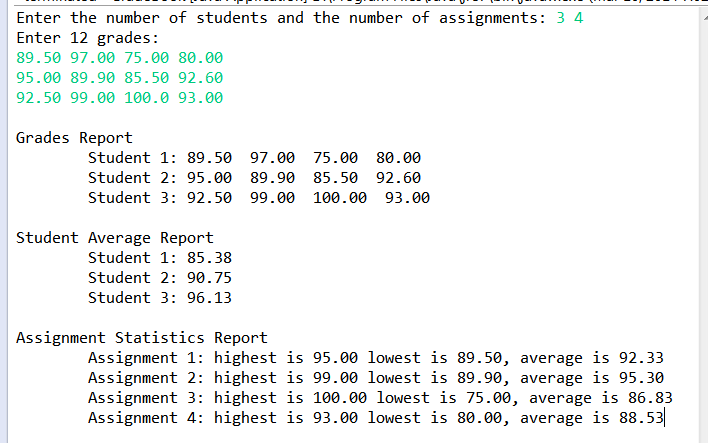
This assignment is to develop a simple grade book application. A grade book has multiple rows and multiple columns. Each column represents an assignment and each row records the grades of a student from those assignments as illustrated below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *3* | *…* | *n* |
| *1* | 89.50 | 97.00 | 75.00 | … | 100.00 |
| *2* | 95.00 | 89.90 | 85.50 | … | 92.60 |
| … | … | … | … | … | … |
| *k* | 92.50 | 99.00 | 100.00 | … | 93.00 |

When the application starts, it should prompt the user to enter the number of students and the number of assignments. After receiving those numbers, the application should prompt the user to enter grades.

After all grades are entered, the application should then display all the grades in the grade book. The grades of each student should be displayed in a single line. After all grades are displayed, the application should produce a report of the average grades of students. In addition, it should generate a report about the statistics of the assignments including the average score, the highest score, and the lowest score of each assignment.

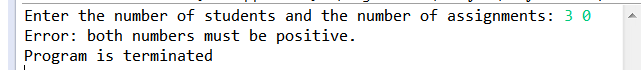
Here is a sample run of three students, four assignments and 12 test case grades:



Assignment Requirements

The following specific requirements must be met:

* The grade book should be implemented as a two-dimensional array
* Create a **sub-method** to read the grades, store the grades in a two-dimensional array (grade book), and then return the grade book to the caller.
* Create a **sub-method** to display all grades in the grade book passed into the method as a parameter
* Create a **sub-method** to compute and display the average grades of the students in the grade book passed into the method as a parameter
* Create a **sub-method** to calculate and report the statistics of all assignments in the grade book passed into the method as a parameter
* The **main method** of the program must call those methods to read the grade book and produce those reports
* If either the number of students or the number of assignments or both is not positive, you program should report the error and terminate the execution as shown in the example below:



* Your program should pass the following test cases:
  1. Negative assignments input
  2. Negative students input
  3. 3 students, 4 assignments, grades matching the test case values above
* Include a screenshot of the sample run (run through **your** program) along with the test cases in your screenshots.

**Submission Requirements**

Your assignment submission should include the following items:

* The Java source code file (i.e., \*.java file)
* The screen captures of the test cases you run

Submit all those files above into the dropbox of this assignment.

**Assignment Assessment**

The assessment rubric for the assignment is below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Items** | **Complete & Correct** | **Incomplete/ Partially Correct** | **Missing** |
| Solution Pseudo-Code (Java comments) (10%) | 100% | By % | 0 |
| Java Program (60%) | 100% | By % | 0 |
| Testing Completeness (30%) | 100% | By % | 0 |