CSCI 1301: Introduction to Computing and Programming

Project 2: Grade Calculator

Introduction

This project will allow you to apply your knowledge of decision statements, variables, assignments, string manipulation expressions, inputs, outputs, algorithm design, compiling, testing, and debugging source code. After completing this project, you'll have your own program that may aid you in calculating your final numerical grade.

You will write a Java application called ***GradeCalculator*** in a file called ***GradeCalculator*.java**. This application will aid a student in calculating his/her current grade in the course as well as finding the average score (s)he will need to achieve a certain letter grade.

The following scale is used to compute the final letter grade in the course (we have omitted +/- letter grades for simplicity):

Letter Grade	Point Range	Meaning
A	90-100	Greater than or equal to 90 and less than or equal to 100
B	80-89	Greater than or equal to 80 and less than 90
C	70-79	Greater than or equal to 70 and less than 80
D	60-69	Greater than or equal to 60 and less than 70
F	Below 60	Less than 60

The final grade for the course will be based on the student's performance in Exam 1, Exam 2, Final Exam, Labs, Quizzes, Projects, and Participation. Your program should prompt the user for the final letter grade (s)he wants to obtain for the course. Then, the program should prompt the user to enter the weight each grading item carries.

Afterwards, the program prompts the user if (s)he knows the score for each grading item in the course. If the user enters "y" or "yes"(ignoring capitalization), the program asks him/her the score for that grading item. Any other response is treated as a "no". The scores for each grading item are out of 100. When prompting the user for the exam scores, the program should not prompt the user for exam 2 or the final exam if the user does not know the score for exam 1. Similarly, if exam 2 is not known then the program does not prompt for the final exam score.

Once the user has entered his/her desired final letter grade, the weights of all grading items and his/her known scores, the program calculates and displays the current grade based on the current scores only using the following formula:

$$currentScore = \sum \frac{weight_{gi} * \ score_{gi}}{totalKnownGradeWeight}$$

where

- ***weight**$_{gi}$*: the weight of grading item *gi*
- ***score**$_{gi}$*: the score achieved in the grading item *gi*
- ***totalKnownGradeWeight***: sum of weights of the grading items of known scores

For example, suppose that the user wants to get an A for the course and (s)he only knows his/her scores for Exam 1 and Exam 2, 92 and 85 respectively. Moreover, Exam 1 carries 15% weight and Exam 2 carries 20% weight. So the user's current score is calculated as:

$$currentScore = \frac{weight_{Exam1} * score_{Exam1} + weight_{Exam2} * score_{Exam2}}{weight_{Exam1} + weight_{Exam2}}$$

Replacing the scores and percent weights by the user's scores and percent weight in the previous formula, the user's current score is:

$$currentScore = \frac{15 * 92 + 20 * 85}{(15 + 20)} = 88$$

*Hint: Dividing integers may give us unexpected results.

Then, the user's current letter grade is determined by checking the range his/her current score falls in the course's grade scale above. The program will display the current score and grade letter for the user. Moreover, if the user has entered a score for all the course grading items, the program should indicate in the message that score and letter grade corresponds to his/her final score and final letter. In the example above, the user's current letter grade is a 'B' according to the course's grading scale.

The formula to calculate the grade average to obtain a final overall score is as follows:

$$avgToFinalLetterGrade = \frac{100 * finalOverallScore - \sum weight_{gi} * score_{gi}}{100 - totalKnownGradeWeight}$$

- ***finalOverallScore***: minimum score in the range of the letter grade the user wants to achieve in the course.
- ***weight**$_{gi}$*: the weight of the grading item ***gi***.
- ***score**$_{gi}$*: the score achieved in the grading item ***gi***.
- ***totalKnownGradeWeight***: sum of weights of grading items with known scores. **Note: if this value is 100 then all grades are known and you should not do the calculation (as it results in a divide by zero).**

The desired letter grade is achievable if the average score for the remaining grading items is not greater than a 100. If the user's desired letter grade is achievable in this semester for the student, the program prints the average score for the student to get that final letter grade. Otherwise, it prints that the user cannot achieve that letter grade for the course.

In our example, the user would like to obtain an A. Therefore, according to the previous formula, (s)he then needs to score a grade average greater than or equal to 91.07692 for the labs, quizzes, projects, participation, and final exam to obtain an A for the course.

Your program must display the grade average with two decimal places. If the computed average score is not exact at the second decimal place, your program should add 0.01 to the average score and then display the result with two decimal places. Alternatively, you could do some math (multiply, divide) and use **Math.floor()** or **Math.ceil()**. In the previous example, the program will display 91.08.

Requirements

1. The name of the class in your Java program must be ***GradeCalculator***. Therefore, the Java source code file must be called ***GradeCalculator.java***.
2. You should try to make your program look just like the examples below when you run your program with the same input.
3. The **input must be specified in the same order** as indicated in the examples. This may seem like a minor detail but variations can cause problems for the graders – especially if they are automating the grading.
4. The desired final letter grade entered by the user should be either A, B, C, D or F (upper or lower case). Otherwise, the program will display an error message and terminate immediately. **Throwing an exception is not an acceptable error message.**
5. If the weights entered for the course grading items do not add up to a 100, your program should display "Weights don't add up to 100, program exiting…" and exit immediately.
6. Your program should be able to process the answers to yes/no questions regardless of the case. Moreover, if the user does not input yes, y, no or n to a yes/no question, the program assumes that the answer is No (example output below).
7. If the user enters a score for all values then you know the final grade. In this case, you should indicate whether or not they received the grade they wanted (example output below).
8. You can safely assume the user will always enter a grade score as an integer between 0 and 100.
9. You can safely assume the user will always enter a non-negative percent weight as an integer between 0 and 100.
10. The current user's score as well as the average score to obtain the user's desired final letter grade must be displayed with **at most two decimal places**.
11. Think about what it means if the **avgToFinalLetterGrade** is a negative value. Handle this case with an appropriate output (it's not an error).
12. Your program's source code must contain the following class comment with the brackets filled in with this project's information (class name, your name, and submission date). The class comment should be placed directly above the class declaration but under the program's import statement(s).

```
/*
* [Class name here].java
* Author:  [Your name here]
* Submission Date:  [Submission date here]
*
* Purpose: A brief paragraph description of the
* program. What does it do?
*
* Statement of Academic Honesty:
*
* The following code represents my own work. I have neither
* received nor given inappropriate assistance. I have not copied
* or modified code from any source other than the course webpage
* or the course textbook. I recognize that any unauthorized
* assistance or plagiarism will be handled in accordance with
* the University of Georgia's Academic Honesty Policy and the
* policies of this course. I recognize that my work is based
* on an assignment created by the Department of Computer
* Science at the University of Georgia. Any publishing
* or posting of source code for this project is strictly
* prohibited unless you have written consent from the Department
* of Computer Science at the University of Georgia.
*/
```

Every Java file of every project you submit **must** have a comment such as this.

Project Submission

- Submit the file ***GradeCalculator*.java** and only that file via eLC.

Project Grading

All projects are graded out of a possible 100 points. Make certain your program compiles before submitting, and thoroughly test your program with different inputs to verify that it is working correctly. All instructions must be followed in order to receive full credit. Read on for a list of additional requirements.

Additional Requirements

These are things that make the graders lives easier, and ultimately, you will see in the real world as well. Remember the teaching staff does not want to touch your code after they gave you requirements; they want to see the perfect results they asked for! Here is a checklist of things you can **lose points** for:

- (100 points) If the source file(s) are not submitted before the specified deadline's late period ends (48 hours after the deadline) or if they do not compile.
- (25-50 points) Late penalties will be deducted as per the course syllabus.
- (10 points) If the source file(s)/class(es) are named incorrectly (case matters!)
- (10 points) If your source file(s) have a package declaration at the top
- (20 points) If the order of your output does not match the examples
- (15 points) If the output text does not match exactly (unless otherwise specified in the lab/project description)

- (100 points) If you use non-standard Java libraries, or other code specifically disallowed by the lab/project
- (10 points) If you are missing your Statement of Academic Honesty
- (10 points) If you have more than one Scanner declared, or declare your Scanner inside of a loop
- If your (10 points) comments or (10 points) variables are "lacking"
    - Here, "lacking" means that you or a TA can find **any** lines of code or variables that take more than 10 seconds to understand, and there is no comment, or the variable name does not make sense (variable names like **`b`**, **`bb`**, **`bbb`**, etc. **will almost never be acceptable)**
- (10 points) Indentation is not consistent throughout your source code
    - Refresh your memory of indentation patterns in chapter 2 in the course textbook
    - Be careful of a combination of tabs and spaces in your files (use one or the other)!

If any of the above do not make sense to you, talk to a TA, or ask on Piazza!

Examples

Your program should work correctly and follow the examples below. The green text is user input. Each example is a separate run of a correctly working program. Please note that some long lines of output are wrapped around multiple lines in this document, and it is okay if your output displays them on a single line. **Do not assume that these examples exhaust all test cases.**

```
----------------------------------------------------------------------
Example 1: weights don't add up
----------------------------------------------------------------------
Grading Scale:
A       90 - 100
B       80 - 89
C       70 - 79
D       60 - 69
F       below 60
What letter grade do you want to achieve for the course? A
Enter Percentage Weights:
Exam 1:         15
Exam 2:         15
Final Exam:     20
Labs:           15
Projects:       20
Participation:  3
Quizzes:        7
Weights don't add up to 100, program exiting...
----------------------------------------------------------------------
Example 2: user has all scores
----------------------------------------------------------------------
Grading Scale:
A       90 - 100
B       80 - 89
```

C 70 - 79
D 60 - 69
F below 60
What letter grade do you want to achieve for the course? A
Enter Percentage Weights:
Exam 1: 15
Exam 2: 20
Final Exam: 20
Labs: 15
Projects: 20
Participation: 3
Quizzes: 7
Enter your scores out of a 100:
Do you know your Exam 1 score? y
Score received on exam 1: 92
Do you know your Exam 2 score? YES
Score received on exam 2: 95
Do you know your Final Exam score? YeS
Score received on final exam: 91
Do you know your lab average? yEs
Average Lab Grade: 92
Do you know your project average? YES
Average Project Grade: 85
Do you know your participation average? YEs
Average Participation Grade: 100
Do you know your quiz average? YeS
Average Quiz Grade: 85
Current Grade Score: 90.75
Your current letter grade is a A
Congratulations! You received the A that you wanted!

Example 3: user has only some scores

Grading Scale:
A 90 - 100
B 80 - 89
C 70 - 79
D 60 - 69
F below 60
What letter grade do you want to achieve for the course? B
Enter Percentage Weights:
Exam 1: 15
Exam 2: 20
Final Exam: 20
Labs: 15
Projects: 20
Participation: 3
Quizzes: 7

Enter your scores out of a 100:
Do you know your Exam 1 score? nO
Do you know your lab average? yes
Average Lab Grade: 50
Do you know your project average? yeS
Average Project Grade: 80
Do you know your participation average? YES
Average Participation Grade: 100
Do you know your quiz average? no
Current Grade Score: 69.74
Your current letter grade is a D
You have to score an average greater than or equal to 86.3 in the remaining grade items to receive an B in the course

---

Example 4: user cannot get the desired grade

Grading Scale:
A 90 - 100
B 80 - 89
C 70 - 79
D 60 - 69
F below 60
What letter grade do you want to achieve for the course? A
Enter Percentage Weights:
Exam 1: 15
Exam 2: 20
Final Exam: 20
Labs: 15
Projects: 20
Participation: 3
Quizzes: 7
Enter your scores out of a 100:
Do you know your Exam 1 score? YeS
Score received on exam 1: 32
Do you know your Exam 2 score? yES
Score received on exam 2: 48
Do you know your Final Exam score? YES
Score received on final exam: 28
Do you know your lab average? yes
Average Lab Grade: 28
Do you know your project average? No
Do you know your participation average? yes
Average Participation Grade: 0
Do you know your quiz average? no
Current Grade Score: 33.16
Your current letter grade is a F
Sorry, you cannot receive an A in the course

Example 5: user enters answers other than yes/no (treated as no)

```
Grading Scale:
A       90 - 100
B       80 - 89
C       70 - 79
D       60 - 69
F       below 60
What letter grade do you want to achieve for the course? C
Enter Percentage Weights:
Exam 1:         15
Exam 2:         20
Final Exam:     20
Labs:           15
Projects:       20
Participation:  5
Quizzes:        5
Enter your scores out of a 100:
Do you know your Exam 1 score? YeS
Score received on exam 1: 45
Do you know your Exam 2 score? nope
Do you know your lab average? kinda
Do you know your project average? Nah
Do you know your participation average? no
Do you know your quiz average? yes
Average Quiz Grade: 70
Current Grade Score: 51.25
Your current letter grade is a F
You have to score an average greater than or equal to 74.69 in the remaining grade items to receive an
C in the course
```

Example 6: user enters an invalid grade

```
Grading Scale:
A       90 - 100
B       80 - 89
C       70 - 79
D       60 - 69
F       below 60
What letter grade do you want to achieve for the course? Z
Input Error
```

Example 7: user achieves the grade they want with grades remaining

---

```
Grading Scale:
A 90 - 100
B 80 - 89
```

C 70 - 79
D 60 - 69
F below 60
What letter grade do you want to achieve for the course? D
Enter Percentage Weights:
Exam 1: 20
Exam 2: 20
Final Exam: 20
Labs: 10
Projects: 10
Participation 10
Quizzes: 10
Enter your scores out of a 100:
Do you know your Exam 1 score? y
Score received on exam 1: 100
Do you know your Exam 2 score? y
Score received on exam 2: 100
Do you know your Final Exam score? y
Score received on final exam: 100
Do you know your lab average? y
Average Lab Grade: 100
Do you know your project average? y
Average Project Grade: 100
Do you know your participation average? n
Do you know your quiz average? n
Current Grade Score: 100.0
Your current letter grade is a A
You will receive at least a D no matter what

You should thoroughly test your program with a lot of different input values to ensure correctness. The above cases do not test all possibilities for error.
