Homework 09 - Grade Calculator

Problem Description

Canvas can be a pain - it can take a long time to log in to on slow connections, and the course grade displayed is not always guaranteed to be correct due to incorrect weight calculations or not-yet-entered grades. This is troublesome! You decide to fix this using JavaFX to provide a GUI (Graphical User Interface) to calculate your course grade.

As the professor mentioned in class, this homework is **optional**, but can replace a lower grade on a previous homework assignment.

Solution Description

For this homework, your task is to build a JavaFX application called `GradeCalculator`, which meets the following requirements:

- Allows the user to enter a grade into a textbox.
- Allows the user to press a button to calculate the grade
- Displays the weighted grade and the corresponding letter grade somewhere in the program
- If a field was left blank or given an invalid input (anything other than a number), the program should indicate somewhere that an invalid input was received.

Your program should contain a text field for each midterm exam, a separate text field for your homework average, and a text field for the final exam. Each box should have a label indicating which catagory the textbox belongs to. Weights for the different assignment types are as follows:

- 57% For the three midterms (19% per exam)
- 18% Programming Homework
- 25% Final Exam

Beyond these requirements, you are allowed large amounts of creative freedom. The layout and inner workings of the program are yours to choose - we just require that the specified functionality be present. Ask on Piazza if you have questions! This assignment may be challenging. We recommend that you utilize the JavaFX API online to find methods and classes that may be useful in accomplishing the tasks laid out for you.

Rubric

- [10] Labels for each text field exist
  - [02] per label
- [10] Text fields exist to allow user input for each assignment
  - [02] per text field
- [10] A button exists in the application
- [10] When the button is pressed, the GUI updates to display the newly calculated information

- [10] Handles the case where one or more fields are empty
- [10] Handles the case where one or more text fields contain invalid input data
- [20] Correctly calculates and displays letter grade
- [20] Correctly calculates and displays weighted average

The Checkstyle cap for this assignment is 30 points. This means that up to thirty points can be lost from Checkstyle errors. We reserve the right to adjust the rubric, but this is typically only done for correcting mistakes. JavaDocs will be required for this assignment.

Allowed Imports

For this assignment, you are offered almost full creative freedom. As such you are allowed to import anything that does not trivialize the assignment. For example, most anything in the `javafx` library is available for you to use in constructing your application. Ask on Piazza if you are unsure. For the most part, though, you will not need to import anything fancy. Refer to Module Twelve - most of the GUI components needed are discussed there. In addition, do not use pre-built themes or GUI files downloaded from the internet, or generated automatically for you - all code must be your own original work!

Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach. For that reason, do not use any of the following in your final submission:

- var (the reserved keyword)
- System.exit()

Collaboration

Collaboration Statement

To ensure that you acknowledge a collaboration and give credit where credit is due, **we require that you place a collaboration statement as a comment at the top of at least one java file that you submit**. That collaboration statement should say either:

I worked on the homework assignment alone, using only course materials.

or

In order to help learn course concepts, I worked on the homework with [give the names of the people you worked with], discussed homework topics and issues with [provide names of people], and/or consulted related material that can be found at [cite any other materials not provided as course materials for CS 1331 that assisted your learning].

Allowed Collaboration

When completing homework assignments for CS1331 you may talk with other students about:

- What general strategies or algorithms you used to solve problems in the assignments

- Parts of the homework you are unsure of and need more explanation
- Online resources that helped you find a solution
- Key course concepts and Java language features used in your solution

You may **not** discuss, show, or share by other means the specifics of your code, including screenshots, file sharing, or showing someone else the code on your computer, or use code shared by others.

Examples of approved/disapproved collaboration:

- **approved**: "Hey, I'm really confused on how we are supposed to implement this part of the homework. What strategies/resources did you use to solve it?"
- **disapproved**: "Hey, it's 10:40 on Thursday… Can I see your code? I won't copy it directly I promise"

In addition to the above rules, note that it is not allowed to upload your code to any sort of public repository. This could be considered an Honor Code violation, even if it is after the homework is due.

Turn-In Procedure

Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- GradeCalculator.java

Make sure you see the message stating "HW09 submitted successfully". From this point, Gradescope will run a basic autograder on your submission as discussed in the next section.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the homework. We will only grade your last submission: be sure to **submit every file each time you resubmit**.

Gradescope Autograder

This assignment will be manually graded, so there will be no autograder provided. Please be careful to compile and test your program, following the instructions in Module Twelve for installing JavaFX and compiling/running programs with JavaFX components.

Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items
- Test your code in addition to the basic checks on Gradescope
- Submit every file each time you resubmit
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points
- Check on Piazza for a note containing all official clarifications