Programming Assignment #1

Grade Calculator

Due: 02/20/2019 at 11:59 PM

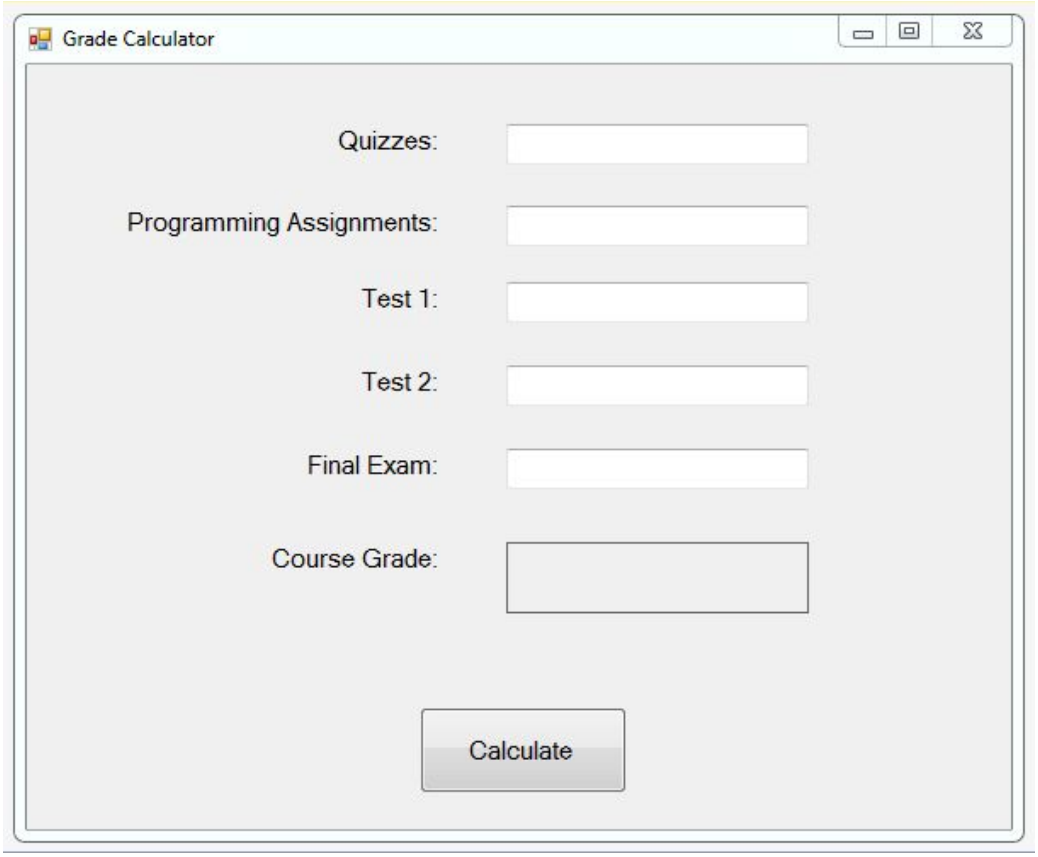
Description

For this assignment, you are going to create a simple final grade calculator application to figure out your grade for this course. You are encouraged to look at the course syllabus, and course website to figure out how your final grade will be calculated.

Note: We are not considering extra credit stuff in this assignment, for simplicity's sake.

User Interface Design (20 pts)

You should try to make your user interface look similar to the following screenshot. You need to have labels and textboxes, as well as a "Calculate" button somewhere in the interface.



Instructions

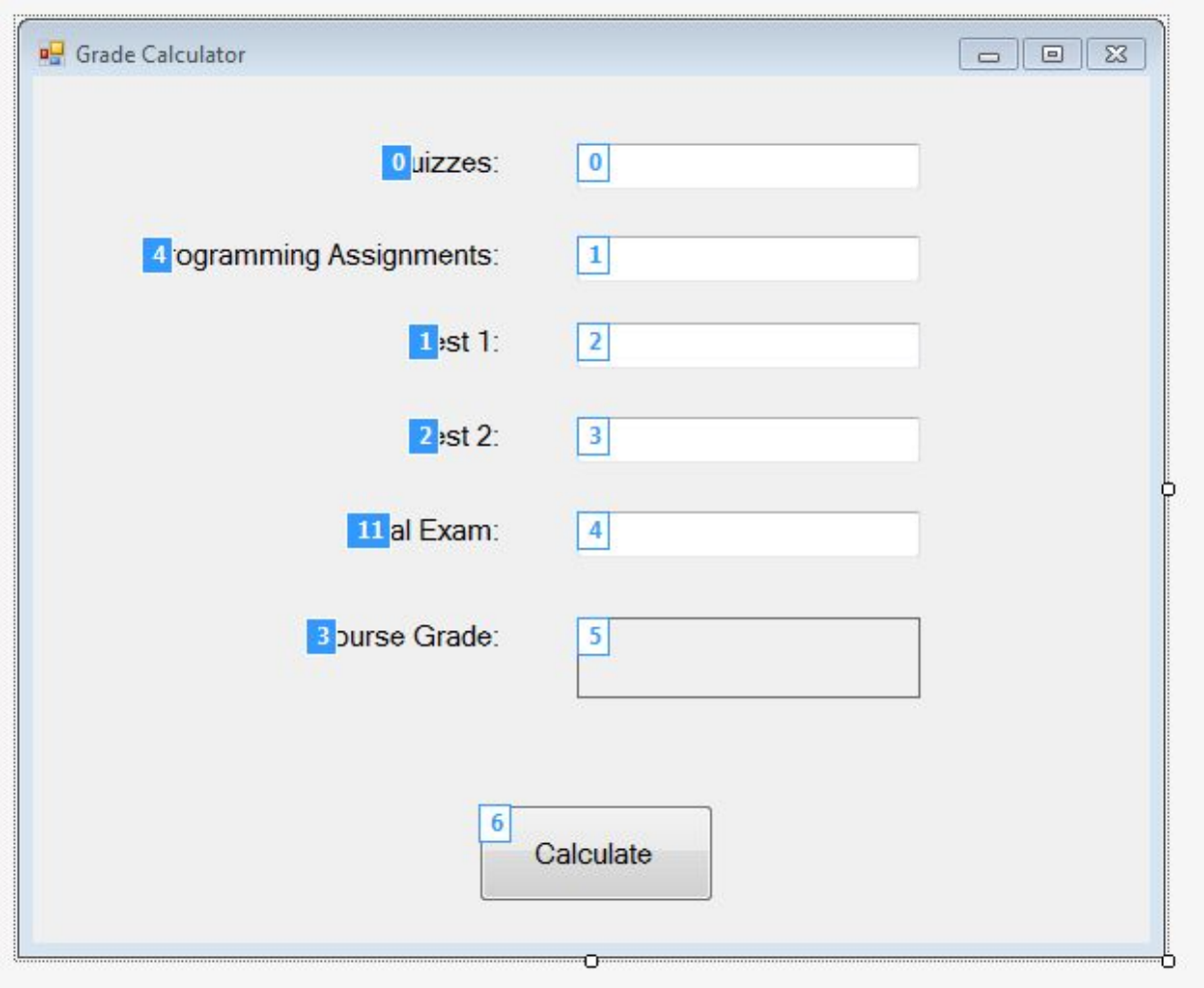
The grading of this assignment has been partially automated by programs that will type in inputs into your program, and will record how your program reacts. Failure to follow directions may result in your program “breaking” the grading scripts, and will result in points being taken off.

Tab Order

You need to make sure your application’s Tab Order is set to go from the topmost “Quiz” text box, all the way down to the “Calculate” button.

This increases the accessibility of your application by allowing users to use only the keyboard to perform tasks. It also ensures the grading scripts will know where to put in numbers while testing your code.

Your tab order for the interface should be identical to the following image for the first seven items:



(You can see this view of your application by clicking on a UI element, then going to “View” > “Tab Order”. Clicking on a UI element will change its Tab Order in this view.)

What this means is that regardless of how those items are arranged visually, hitting the Tab key should cause the cursor to jump between interface elements in the following order:

1. Quizzes
2. Programming Assignments
3. Test 1
4. Test 2
5. Final Exam
6. Calculate

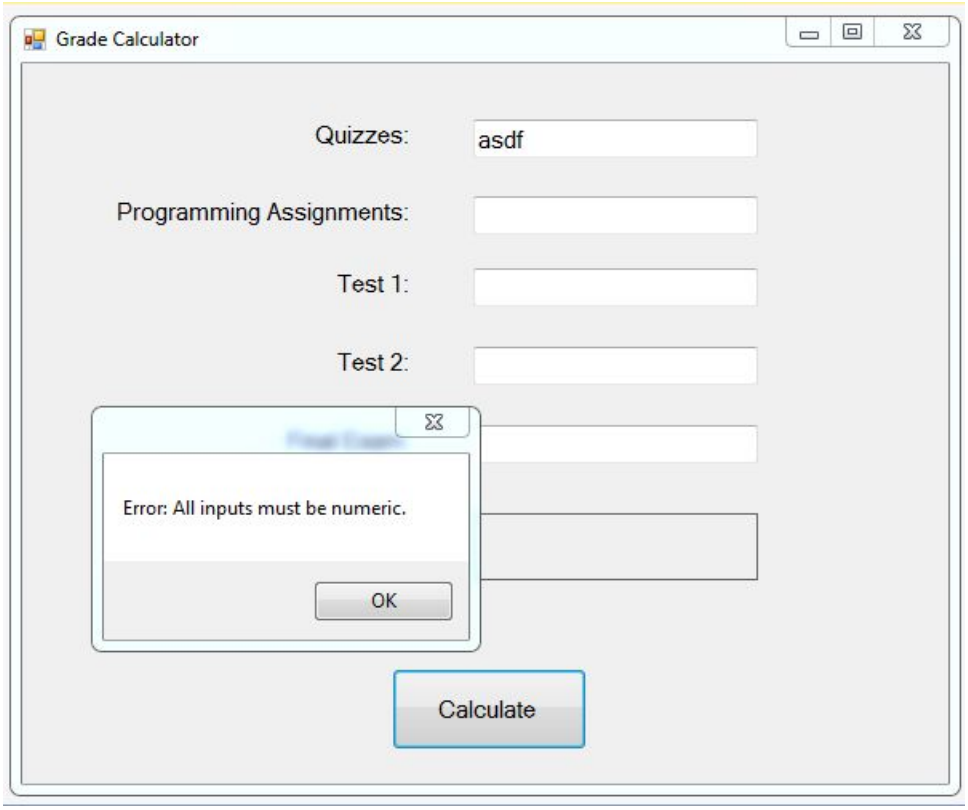
Computing the Grade (50 pts)

Your program needs to put the correct final grade in the UI element next to “Course Grade”. To figure out the weighting of each graded category, please consult the syllabus and/or course webpage.

Remember: The final grade shown is out of 100%, so remember to try putting in all 100’s to make sure you haven’t over-weighted a category!

Handling Errors (10 pts)

We’ve seen in class how to handle bad inputs by the user. For this assignment, **you need to handle errors by displaying a popup message** to the user, as shown below:



The message for bad inputs can be pretty generic, like the message shown in the image above. The important thing is that you inform the user about what went wrong in a useful and concise way.

Your program will be tested by the grading scripts to see how robustly it can handle bad inputs from the user. Test your program carefully!

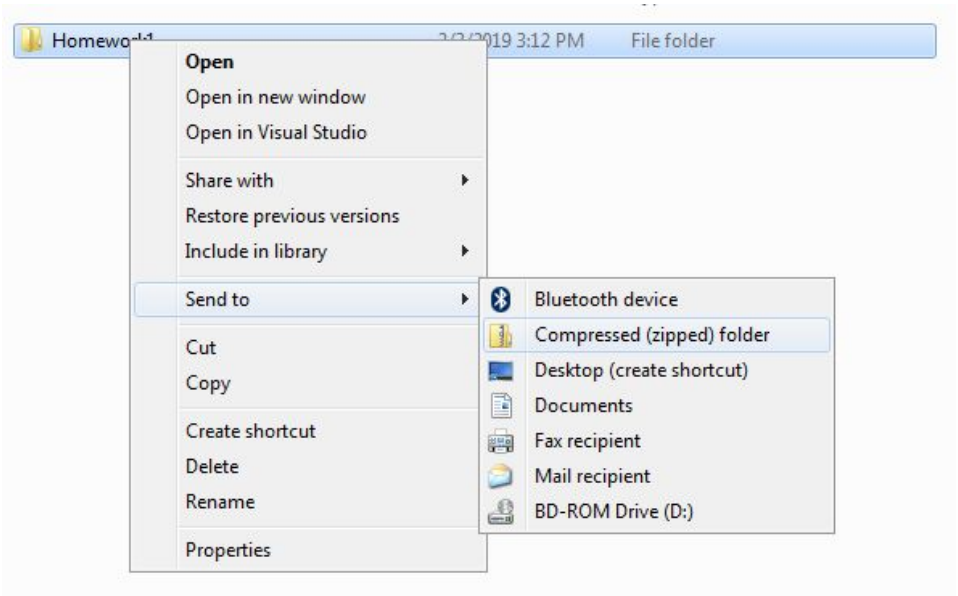
Coding Style (20 pts)

The Instructor/Grader will look at your source code for how you implement your grade calculations. We will be looking for the following things:

- **Your name in a comment** at the top of the file.
- Clear and concise comments, where appropriate.
- Clear/Informative variable names.
- Good layout of the code (sometimes a blank line in the right spot makes the code look better!)

How to turn in your assignment

1. Find the `C:\Users\<YOUR_USERNAME>\source\repos` folder.
2. Right click on the folder where your Visual Basic project lives.
3. Select “Send to” > “Compressed (zipped) folder” to create a ZIP file of your project. The menu should look something like the following picture:



4. Upload the ZIP file to Blackboard.

Grade Rubric

- UI Design/Layout :: 20 pts (Includes Tab Order requirements)
- Code Style :: 20 pts
- Correct Results :: 50 pts
- Error-handling :: 10 pts

Total :: 100 pts
Extra Credit :: +10 pts

Extra Credit (Optional):
Handling numbers outside the 0-100 range (+10 pts)

Depending on how you check for bad inputs from the user, your code may not catch when users type in valid numbers outside the 0-100 range. Ideally, **we'd like to display a pop-up message to the user** that they've made a mistake somewhere.

Examples of valid numbers outside the 0-100 range:
-45
110
11111111111111111111111111129
4000
-347
-1
101

Try putting some of those numbers into your program, and see if your code can detect the mistake.

If your code is able to inform the user when they type in valid numbers outside the 0-100 range, you can earn up to 10 points of extra credit on this assignment.