<u>Homework 3 (5 points)</u> This homework is due in your blog by Oct 9, 6AM

Please start from the Homework 3 starter files included in the Assignments area. This version contains some additional advice and hints in the comments as well as comments designating where to insert the homework code. Each required question is worth 1 point, graded in 1/4-point increments unless otherwise specified.

1. Add an additional button to the **index.html** file. It should appear after the two "Add" buttons that we created in the workshop, and it should have a text label of "Update Grade". For coding purposes, you can identify the button however you want. Then go to **homework3-start.js** and add code to connect this new button to a function called that will prompt the user for grade information.

2. Write a function that prompts the user for three pieces of information using three dialog boxes: a student ID number, an assignment ID number, and the number of points that the student earned on the assignment. Be sure to validate this information and not go on to the next dialog box until the user has entered valid information. Note that in this case all three inputs should be numbers, and those numbers can include 0. However, you can be assured that there will be no student ID or assignment ID below 0 (since Arrays index starting at 0), and we will also assume students can't earn negative points on an assignment. Also, we won't bother checking to see if the user enters too high a number for now. At the end of this function, you should call the function called **updateStudentGrade**, which I have started for you, sending it the data you've just collected as the parameters.

3. I have already written most of the code for **updateStudentGrade** for you. The one fragment of code that you have to write is a loop to make **targetColumn** change what it's looking at from the "Overall" (i.e., Percentage) column to the appropriate Assignment ID. Look at the **nextSibling** property of the **targetColumn** property that I've set up for you. Could you perhaps update the **targetColumn** to be the sibling of the current **targetColumn** and keep doing that until you reach your goal? After you've used a loop to count over to the correct column, then set that column's text content to the number of points. **CAUTION**: Check your work carefully on this one, as you could easily end up off by one column if you're not careful! The test cases at the end of the program should now update **Homework #1** for **Adam Anders** to 5 points and **Homework #2** for **Beth Booker** to 10 points.

4. I have already written most of the **updateGrades** function that actually calculates changes to the student's percentage grade for you. This function needs to be called whenever you add or update any rows or columns on the spreadsheet. Assume that even adding a new student requires updating the student's "Overall" grade column, since perhaps in the future we will add the ability to add students along with some grades (though not this week)! You should add a proper call to the **updateGrades** function as the final statement within *exactly three* functions in the program (I haven't marked the locations for this problem, since the challenge is to figure out where the calls are needed. You will earn 1/3 of a point for each correct location (up to 1 point for this question) and you will lose 1/3 of a point for each incorrect location (down to 0 points for this question)).

5. The only part of **updateGrades** that I haven't written for you are the lines that perform the actual update to the DOM. Using the data that is available to you in the function, figure out a way to access the "Overall" (i.e., Percentage) **td** for the current student (remember, you are looping through all of the students here) and change its text to match the value that has been calculated. The format for the text value should be the student's percentage grade rounded to 1/10th of a percent with a (e.g., "90.3%"). Make sure to add the "%" sign to the end as well!

BONUS PROBLEMS (1/2 POINT EACH)

1. Change the prompts in your grade updating function so that the user is told the range of student ID numbers (e.g., "0-2" if you have three students) and assignment ID numbers. Prompt the user to re-enter the data if the number entered is less than 0 or greater than the current maximum number for student ID or assignment.

2. Move the calls to **addStudentRow** and **addGradeColumn** out of the two constructor functions (**Student** and **Assignment**) to a place that works equally well without mixing up data elements and visual elements as much as the code currently does. (HINT: You have a couple of functions that "create" things, so maybe you could use those in anticipation of later on writing similar functions to "delete" things?). In addition to moving and rewriting the function calls, you will also have to figure out a way of setting the **studentRow** property for the **Student** objects after **addStudentRow** has been called from its new location. You will have do something similar to set the **gradeColumn** property on the **Assignment** objects within **assignmentData** as well.