**Total: 40 points**

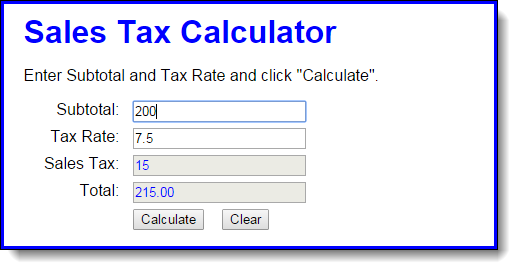
**All assignments should be individual work. Discussion with class mates is allowed but the final code has to be written individually. If you take code from any Web site, provide the links in your submission files.**

**All the files you need for this exercise are available on Blackboard->Course Schedule->A2.zip**

***Start early.***

***DO NOT USE jQuery or CSS to implement the programs. You can use CSS only for styling purposes. Solutions should be general enough so that if the size or content of arrays changes, your code will still work.***

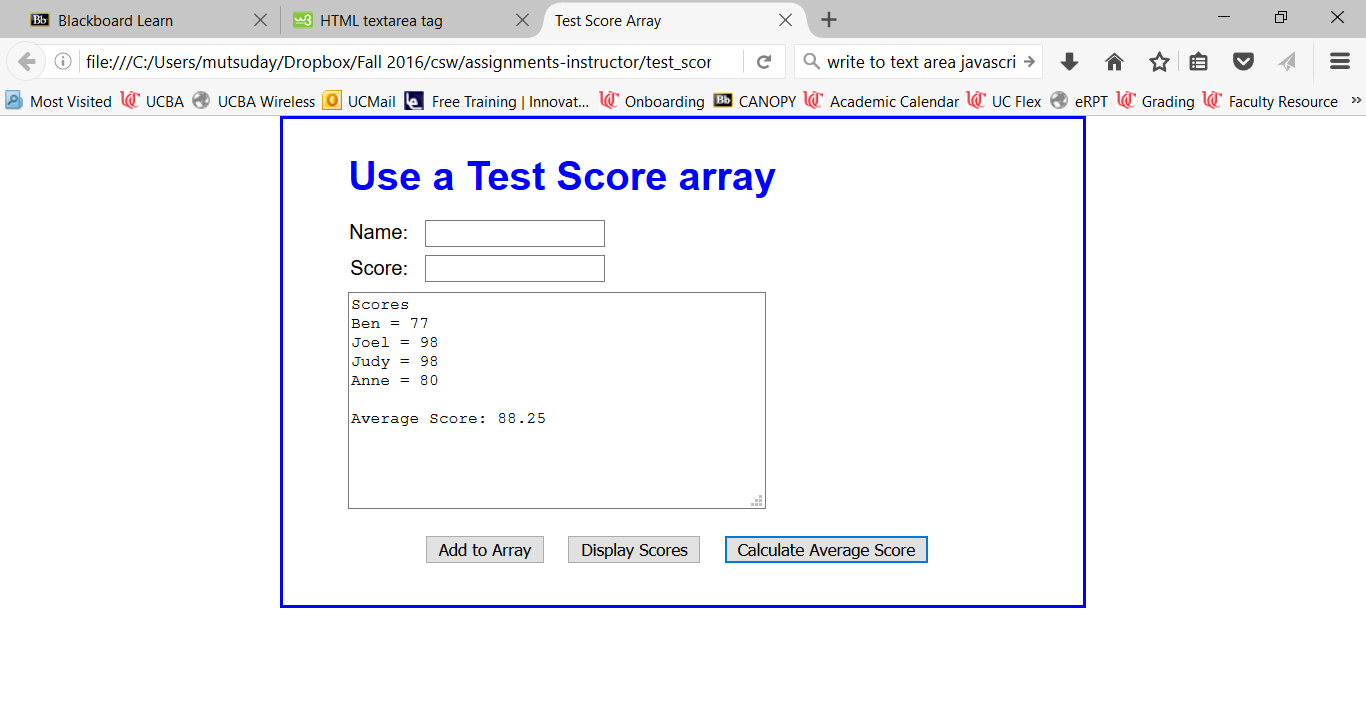
1. (15) In this exercise, you’ll develop an application that calculates the sales tax and invoice total after the user enters the subtotal and tax rate.



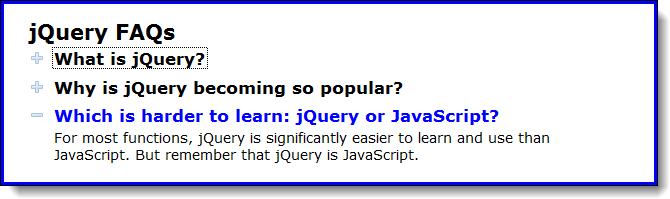
1. Open the HTML and JavaScript files provided in the folder sales\_tax
2. Then, run the application to see the user interface shown above, although that interface won’t do anything until you develop the JavaScript for it.
3. In the JavaScript file, write the code for calculating and displaying the sales tax and invoice total when the user clicks on the Calculate button.
4. Add the JavaScript event handler for the click event of the Clear button. This should clear all text boxes. Add JavaScript event handlers for the click events of the Subtotal and Tax Rate text boxes. Each handler should clear the data from the text box when you click on them to type.
5. If you haven’t already done so, add data validation to this application. The subtotal entry should be a valid, positive number that’s less than 10,000. The tax rate should be a valid, positive number that’s less than 12. The error messages should be displayed with span tags, and the error messages should be:

Must be a positive number less than 10,000  
Must be a positive number less than 12

1. Add JavaScript code that focuses the cursor to the Subtotal field when the application starts and when the user clicks on the Calculate button.
2. (15) In this exercise, you start with the declarations for two arrays, a names array and a scores array. Then you will add one button that will display the scores and another button that will calculate and display the average score.



1. Open the HTML and JavaScript files provided in the folder *test\_scores*
2. If you run the application, you can enter a name and score and then click the *Add to Array* button to add names and scores to the arrays, but you can’t use the other buttons.
3. Look at the JavaScript code to see that it starts with the declarations for two arrays that contain four elements each. The first is an array of names. The second is an array of scores. The JavaScript code also includes the $ function, an onload event handler, and the addScore() function for adding a name and score to the arrays.
4. Add an event handler for the Display Scores button that displays the names and scores in the arrays, as shown above. Remember to attach this handler to the button in the onload event handler. After you test this with the starting array values that are shown above, add a name and score to the arrays and test the display again. WK2 worksheet will be very helpful here.
5. Add an event handler for the Calculate Average Score button that calculates and displays the average of the scores in the array. Keep any text that are already on display. Test this after you have added a name and score to the arrays. WK2 worksheet will be very helpful here.
6. Modify the event handler for the Add to Array button so the data in the text area is cleared after a name and score are added to the arrays.
7. (10) This exercise requires you to make a minor modification to the FAQs application in chapter 3. When you’re done, this application should work the same as before, except that only one answer can be displayed at a time. In other words, when the user clicks on a heading to display the answer, the other answers must be hidden. This takes just a few more lines of code, but you have to think it through.



* 1. Open the HTML and JavaScript files provided in the faqs folder. Then, run the application to refresh your memory about how it works.
  2. Add the necessary code in the JavaScript file so only one answer can be displayed at a time.

**Submission**  
Upload the three folders as a single .zip file to Blackboard under Submit Assignments ->Assignment 2. Name your file as ***YourUsername\_A2.zip***  
e.g. If I were to upload it, it would be mutsuday\_A2.zip

Test your code on a wide range of inputs. Some exercises show some sample inputs and the expected results. Use these examples as a starting point. Your file must successfully execute in order to be considered for a grade- absolutely no syntax errors. Comment out any code that does not work for partial credit.