Assignment 5

(Use JavaScript to access and update HTML elements, decision and loop)

(Total: 100 points)

Learning Objectives

- Hands-on practice on how to access and add HTML elements, and how to add and change attributes for selected elements.
- Hands-on practice on how to use array, how to create combined string, how to use logical operators, if/else statements, and for loop to set up decision making and loops in JavaScript.

Complete your assignment

Download the zip file named "**CS381_A5.zip**" and then unzip the zip file. In the unzipped folder (by default with name "CS381_A5"), you will find two files: **testScoreV2.html**, **testScoreV2.js**, and **score.css**. Add JavaScript code in the "**testScoreV2.js**" to allow users to choose how many tests to consider on the web page, and then display test score input boxes on the web page to allow users to input test scores and then display total score, average score, and final letter grade. The following are the detailed steps to complete this assignment:

Step 1. [40 points] In "**testScoreV2.js**", complete **setupInputBox()** function to set test input boxes on web page according to the number of tests that user wants to consider.

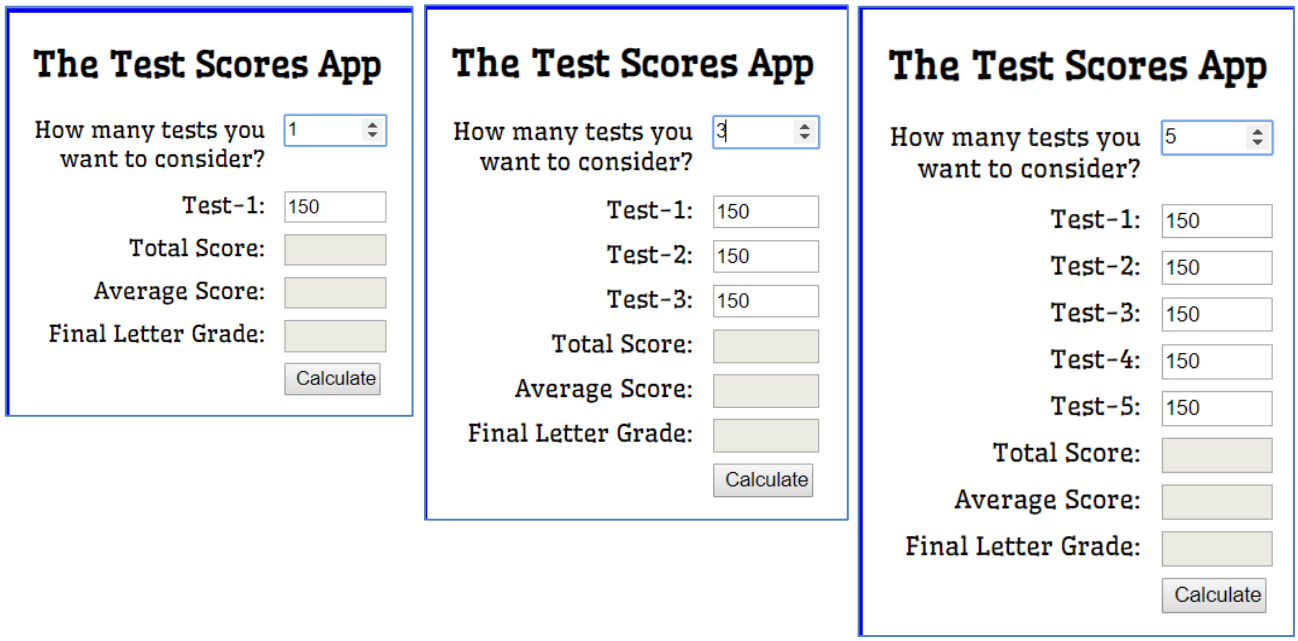
(1) [10pts] Add a condition in the **if()** statement so that if the number of tests that user select is a valid numerical value within the range: **1 to 5**, then new test input boxes will be added by adding new element and test nodes to the DOM tree using a ***for*** loop.

(2) [10pts] Create individual element node, and text node. In the **for** loop, for each iteration, create a <label> element object, a <input> element object, and a
 element object, and a text node object for the <label> element object. (You may use *createElement()* method, and *createTextNode()* method.)

(3) [10pts] Add attributes to those new created element nodes. Add a ***for*** attribute to new <label> element created in part(2), and add ***id, type,*** and ***value*** attributes to new <input> element created in part(2), so that those new elements will have exactly the same structure as the existing <label>, and <input> element in HTML. (You may use *setAttribute()* method.)

(4) [10pts] append new <label>, <input>, and
 elements to the element with *id="testInputs"* by using *appendChild()* method.

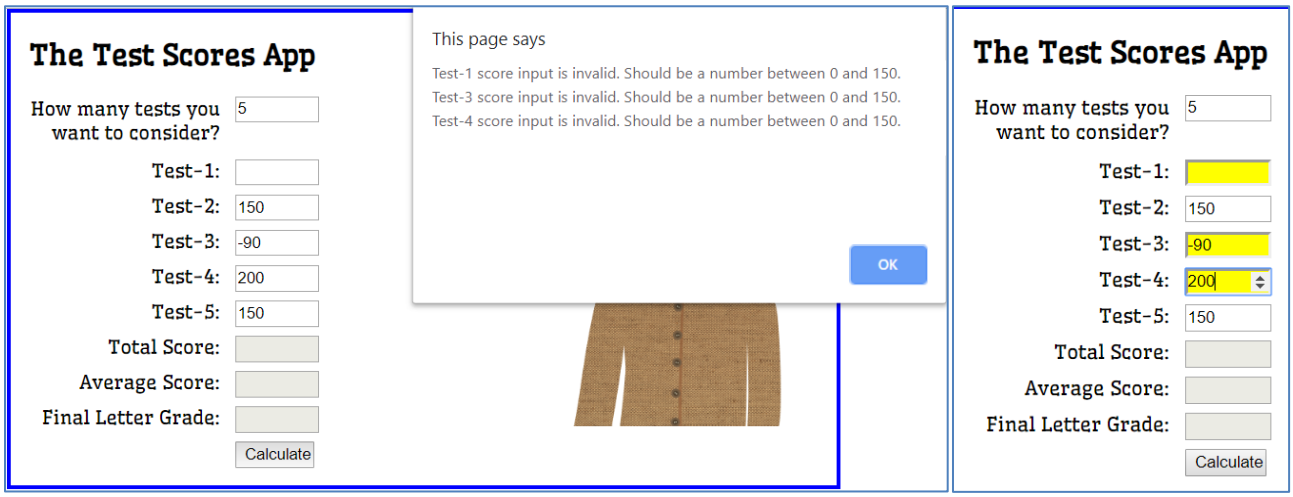
For example, if number of tests that user select is 3 on the web page, then executing thefunction will add three <label>, three <input>, and three
 element nodes to the DOM tree. If number of tests that user select is 5, then executing the **setupInputBox()** function will generate five <label>, five <input>, and five
 elements and add them to the DOM tree.



Step 2. [40 points] In ***processEntries()*** **function,**

(1) [10pts] in the ***for*** loop, add js code to read in user inputted test score(s) from input test score boxes on the web page.

(2) [15pts] in the ***for*** loop, add js code to validate each test score to make sure all inputted test scores are **numerical values between 0 and 150** (including 150). If an inputted test score is invalid, generate an error message, and add that error message to the message variable (a string variable defined in the function), and highlight the input boxes in which there are invalid inputs (by changing the value of class attribute to "error") . If an input test score is valid, store that input test score into score array (an array variable defined in the function).



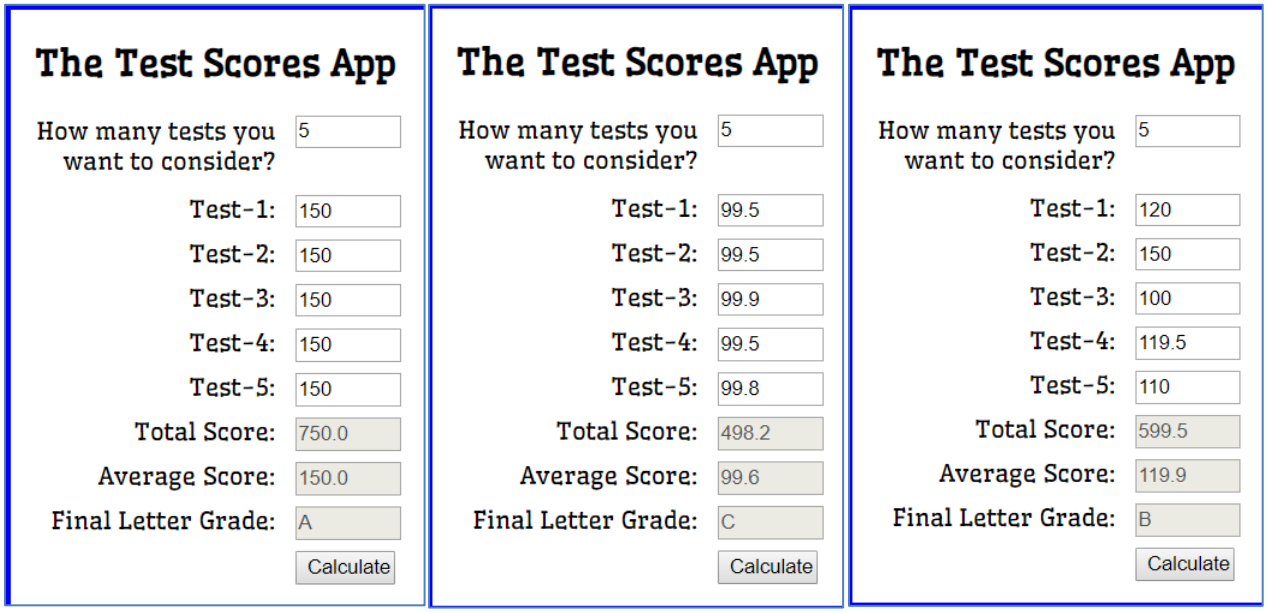
(3) [15pts] add js so that when all inputted test scores are valid, compute total score (with one decimal place), average score (with one decimal place), and final letter grade, and display them in the input boxes in the element with id='result' on the web page. If not all inputted test scores are valid, then create an alert box to display an error message similar as shown in the figure above.

Note: You may define and use extra variables, functions, etc. as you needed to complete above steps.

Final letter grade is determined based on the grading scheme as show below.

Average Score	Final Letter Grade
150 and >=120	A
<120 and >=100	B
< 100 and >=80	C
<80 and >=60	D
Below 60	F

(*See in Lecture 4.1 and sample code about using loops, and if-else statements in JavaScript.***)**



Step 3. [15 points] Test and debug your program (you may use the developer tool in browsers help you) to make sure there is no syntax and logical errors. If there are some errors that you cannot figure out, please write a notes in your program or leave a message in your dropbox.

The data files (CS381_A5.zip) needed for this assignment can be found in the Assignment-5 dropbox on Canvas Course Site.

Submit your assignment [5 points]

After you complete, please **save and zip** the "CS381_A5" folder that including all the files, and then rename and submit the zipped file "**A5_yourLastNameFirstName.zip**", to the assignment drop box named **Assignment-5**.

Note: If your computer does not have the ability to "zip" (i.e., compress or uncompress) files, download and use a free zip program such as 7-Zip (http://www.7-zip.org/)