CE212 ASSIGNMENT 1 2021

Set by: Mike Sanderson

Credit: 10% of total module mark

Deadline: 11.59.59, Wednesday 17 February

Introduction

The assignment comprises three independent exercises. They are not equally weighted.

You should refer to the Undergraduate Students' Handbook for details of the University policy regarding late submission and plagiarism; the work handed in must be entirely your own.

It is expected that marking of the assignments will be completed by the beginning of week 24.

You may, if you wish, use JQuery, but are not required to do so.

Submission and Marking Criteria

Copies of all code (including CSS, JavaScript and HTML files) must be submitted to FASER by the deadline in a single `.zip` or `.7z` file. (Marks may be deducted for files submitted in any other format.)

Other than where specific instructions have been given about coding (e.g. the use of an array of objects in exercise 2) the only criteria to be used for marking of this assignment are that the web pages have the correct functionality, and, where appropriate, the required style.

CE212 ASSIGNMENT 1 2021

Exercise 1 – Dynamic Data Table [30%]

The requirement for this exercise is to use JavaScript (together with CSS and HTML) to produce a dynamic table of data listing conversion temperatures in degrees Celsius and degrees Fahrenheit. The web page must obtain from the user the range (in Fahrenheit or Celsius) of the distances to be displayed and produce a table showing each temperature in the range alongside its equivalent in the other unit, which must be displayed to ***exactly one decimal place***. The user must be able to select whether the conversion is from Celsius to Fahrenheit or from Fahrenheit to Celsius. The JavaScript may be invoked by either a button-click of a menu selection.

Table header cells must have a background colour that differ from data calls; alternative rows must have different text colours.

Text box input fields should be used for the entry values for the start and end of the range. The user must supply at least one value; he or she may leave the second field empty.

Validation must be performed – the first value must be an integer and the other (if supplied) must also be an integer. An alert or error message should be displayed if the validation fails. (If the user enters a number that is not an integer you may either reject it or warn the user that the input has been truncated to an integer). The user may enter the two values in either order but the program should always display the table in ascending order. If only the first value is you should use 0 for the other value.

CE212 ASSIGNMENT 1 2021

Exercise 2 – Marking Form [30%]

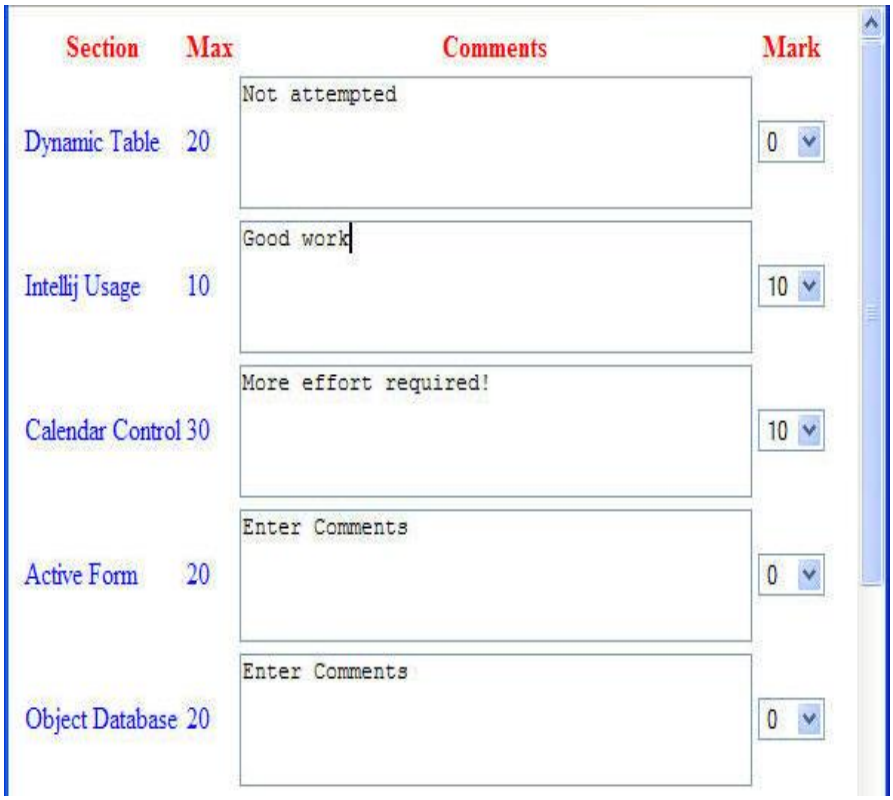
This exercise requires the creation of a marking form for an assignment.

Your solution should have components as shown below and also display the total mark, which should be updated as marks are entered.

The form must be created using JavaScript. The details of sections (i.e. their names and maximum marks) must be stored in an array of objects (which must not themselves be arrays) so that it is easy to change these details by changing the initialisation of the array in one place in the JavaScript file; all of the HTML-generation code must obtain the necessary data from the array. You should ***not*** assume that the array will always contain details of exactly five sections or that the maximum marks will always total 100.

Each drop-down menu should contain all of the integers from 0 to the maximum mark for the section.

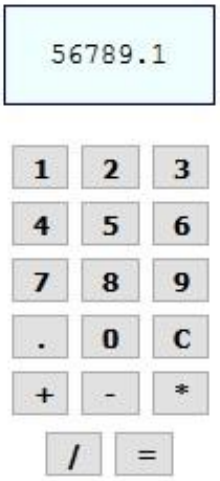
The form should have a Clear button, which when clicked, should clear all of the comments and reset all marks to 0.



CE212 ASSIGNMENT 1 2021

Exercise 3 - Calculator [40%]

The file **`calc.html`**, available on Moodle, generates the layout of a simple calculator as seen below. This should be used as the basis for the writing of a fully-functional calculator.



You will need to write JavaScript functions to respond to button presses. When a digit button is initially pressed or is pressed after the use of an operator button or the **=** button or the **C** button the currently-displayed number should be cleared and the digit shown; subsequent presses of digit buttons or the **.** button should append to the display. (The **.** button should be ignored if the displayed number already contains a decimal point or if the number currently displayed is the result of a calculation; numbers such as .35 must be entered as 0.35)

When an operator button is pressed the displayed number and the selected operator should be stored; subsequent presses of operator button or the **=** button should cause a calculation to be performed using the stored number, the stored operator and the current number; its result should be displayed. If the button was an operator, the new number and operator should be stored.

When the C button is pressed the display should be reset to 0 and any stored number and operator should be cleared.

CE212 ASSIGNMENT 1 2021

The following sequence shows what should happen when the display initially stores 0 and the user wishes to calculate 3.5+2.1*4. (Note that all calculations should be performed from left to right; there must be no operator precedence rules.)

3 pressed: display becomes 3
. pressed: display becomes 3.
5 pressed: display becomes 3.5
+ pressed: 3.5 and + are stored; display is still 3.5
2 pressed: display becomes 2
. pressed: display becomes 2.
1 pressed: display becomes 2.1
***** pressed: 3.5+2.1 is evaluated; display becomes 5.6; 5.6 and * are stored
4 pressed: display becomes 4
= pressed: 5.6*4 is evaluated; display becomes22.4; store is cleared

You may make any reasonable assumptions about precision of calculations such as 1/3 and about what will happen if the number will not fit on the display.

Improve the display by adding CSS so that number buttons and operator buttons have different colours, etc. You may make any changes you wish to the layout and, if you wish, add extra buttons, but the calculator must have the 17 buttons as seen in the diagram.

[About 35 of the 40 marks for this exercise will be awarded for functionality and the remaining 5 for presentation.]