



UOW COLLEGE AUSTRALIA

PATHWAYS TO UNIVERSITY OF WOLLONGONG

## DPIT128: Introduction to Web Technology (Spring 2023)

**Assignment 4 (15%) due by Friday 1st of September 2023 at 11:55 PM**

### Learning Outcomes

- Ability to incorporate JavaScript code into CSS enhanced HTML documents
- Experience using a variety of JavaScript features (e.g. arrays, objects, timers, randomisation, local storage etc)
- Ability to utilise the various control structures available in JavaScript to perform decisions and repetitions

### Submission Instructions

You will need to submit a total of 6 files using the “Assignment 4” submission facility on Moodle. Please see the table below for an overview of the files (including their names and extensions) that are required for each task.

Task	Files Required
1	A4_TASK1.html
2	A4_TASK2.html
3	A4_TASK3.html
4	A4_TASK4.html
5	A4_TASK5.html
6	A4_TASK6.html

Assignments are marked using the **Google Chrome** browser. Check all work in this same browser prior to submission.

### Please be aware of the following:

1. Late submissions will incur an awarded mark deduction of 10% for each day late.
2. Submissions more than 3 days late will not be marked (a zero will be automatically awarded).
3. Please apply for academic consideration if you need an extension. The UOW College “Academic Consideration Policy” can be found at the following URL:  
<https://www.uowcollege.edu.au/about/governance/policies-procedures/>
4. Plagiarism is treated seriously! If we suspect that work has been copied, ALL students involved are likely to be awarded a mark of zero for the entire assessment.
5. This assignment is scheduled to be marked during Weeks 10 & 11.

There are 6 tasks in this assignment.

**Task 1:** Create the HTML and CSS code needed to reproduce the following web application interface. Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown below.



**Design Requirements:**

- The heading text should be displayed within an `h2` element.
- All images must have their border specified within the same CSS rule (see “**Design Resources**”).
- The above rule should also adjust the width of each `img` element to 110px.
- A border radius effect with a shorthand value of `5px 15px` must be applied to each image.
- The text featured directly beneath the images should be displayed within an `h3` element.
- The `input` element that follows must have its alignment set by a rule with an ‘*attribute*’ selector.
- The above-mentioned rule should also adjust the width of the `input` element to 500px.
- A placeholder attribute must be included where appropriate.
- Double-clicking the first image should display a JavaScript prompt (see [Figure 1](#)).
- The value entered into the prompt must then be displayed within a JavaScript alert (see [Figure 2](#)).
- Moving the mouse over the second image should alter the text contained within the `h3` element.
- The altered text must match the output displayed within the “Final State” image (see [Figure 3](#)).
- Scrolling the mouse wheel over the third image should display a text value inside the `input` element.
- The text value must also match the output shown within the “Final State” image (see [Figure 3](#)).

**Design Resources:**

HTML5 Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/html_5.png">https://mywebspace.net.au/UOWC128/assignment_4/images/html_5.png</a>
CSS3 Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/css_3.png">https://mywebspace.net.au/UOWC128/assignment_4/images/css_3.png</a>
JS Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/java_script.png">https://mywebspace.net.au/UOWC128/assignment_4/images/java_script.png</a>

Border Effect	1px dotted <a href="#">#808080</a>
---------------	------------------------------------

Element Style Parameters (h3)		Element Style Parameters (input)	
Colour	<a href="#">#009900</a>	Colour	<a href="#">#ff0000</a>
Font Family	courier	Font Family	arial
Text Shadow	3px 3px 3px <a href="#">#a6a6a6</a>	Font Style	italic

**Please Note:**

- HTML tables must **not** be used to layout this application interface.
- Save your completed file as “A4\_TASK1.html”.
- This will be the first of 6 *separate files* that you must submit for this assignment.

**This task is worth 2 marks.**

**Task 2:** Create the HTML and CSS code needed to reproduce [this web application interface](#) (see [Figure 1](#)). Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown within the interface imagery.

**Design Requirements:**

- The heading text should be displayed within an `h2` element.
- The image must have a width of 200px and a rounding effect set to 100px.
- A matching CSS shadow effect should also be applied (see “**Design Resources**”).
- The three CSS declarations mentioned above must be placed within the same rule.
- A `label` element should be used with the text description for each radio button and checkbox.
- The radio buttons must be grouped with the first/topmost selected by default.
- The checkbox for “Reverse Direction” should be unchecked by default.
- The following actions must occur when the “Start” button is clicked:
  1. Any existing/active JavaScript timers should be *stopped*
  2. A single JavaScript timer must be *created* and *activated*
  3. All radio buttons should be *disabled* to prevent any changes while the timer is active
- The timer must update the clock image using the interval specified by the selected radio button.
- The clock images should be shown in the order listed below (see “**Design Resources**”).
- The display order of these images must be reversed whenever the checkbox is checked.
- The following actions should occur when the “Stop” button is clicked:
  1. Any existing/active JavaScript timers must be *stopped*
  2. All radio buttons should be *enabled* to allow the selection of a new interval
- A “pointer” CSS cursor effect must be applied to all `label` elements and interface buttons.
- Matching background colours should be applied where appropriate (see “**Design Resources**”).
- A suitable border and `arial` font must also be included (see “**Design Resources**”).
- An array should be used to store the various image paths within your JavaScript code.

### Design Resources:

Shadow Effect	5px 5px 5px 5px <a href="#">#888888</a>
Background 1	<a href="#">#f8f8f8</a>
Background 2	<a href="#">#bfbfbf</a>
Border Effect	1px dotted <a href="#">#808080</a>

Clock Image 1	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_1.png">https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_1.png</a>
Clock Image 2	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_2.png">https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_2.png</a>
Clock Image 3	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_3.png">https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_3.png</a>
Clock Image 4	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_4.png">https://mywebspace.net.au/UOWC128/assignment_4/images/clock_state_4.png</a>

### Please Note:

- Your solution must use *more than one* user-defined JavaScript function.
- HTML tables should **not** be used to layout this application interface.
- Save your completed file as “A4\_TASK2.html”.
- This will be the second of 6 separate files that you must submit for this assignment.

This task is worth 2 marks.

**Task 3:** Create the HTML and CSS code needed to reproduce [this web application interface](#) (see [Figure 1](#)). Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown within the interface imagery.

### Design Requirements:

- The heading text should be displayed within an `h2` element.
- An image must be used to show the current state/status of the game (see “**Design Resources**”).
- The three `input` elements featured directly beneath the image should:
  1. Contain a width of 110px and height of 80px along with a font size of 50px
  2. Show a matching border, colouration (see “**Design Resources**”), alignment and bold styling
  3. Include a border radius of 25px and a margin of 20px
  4. Be set to *read only* so that any displayed values cannot be altered by the user
- A `label` element must be used with the text description for the “Enter a number...” field.
- An `arial-16px` font along with a line height of 23px should be applied to the abovementioned label.
- The user’s input must be collected within an `input` element that:
  1. Contains a width of 150px and height of 80px along with a font size of 50px
  2. Show a matching border, colouration (see “**Design Resources**”), alignment and bold styling
  3. Include a border radius of 5px, a margin of 10px and an `autofocus` attribute

- A button labelled “Start” should be included and styled as follows:
  1. Contains a width of 100px and a padding of 10px
  2. Shows a matching border and colouration (see “**Design Resources**”)
  3. Includes a border radius of 5px
- The following actions must occur when the “Start” button is clicked:
  1. The user’s input is validated

**If the input is valid:**

  2. Three unique random numbers between 1 and 10 (*inclusive*) are generated
  3. Each number is displayed within one of the three `input` elements (*1 number per element*)
  4. A check is performed to see if the user’s input matches any of the unique random numbers
  5. The image is altered to indicate the result (see [Figure 2](#) and [Figure 3](#); see “**Design Resources**”)
  6. An audio file is played if a match is found (see “**Design Resources**”)

**If the input is invalid:**

  2. The image is altered to indicate the result (see [Figure 4](#); see “**Design Resources**”)
  3. No subsequent actions are taken
- A “pointer” CSS cursor effect should be applied to all `label` elements and interface buttons.

#### Design Resources:

Initial Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/initial_state.png">https://mywebspace.net.au/UOWC128/assignment_4/images/initial_state.png</a>
Win Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/win_state.png">https://mywebspace.net.au/UOWC128/assignment_4/images/win_state.png</a>
Loss Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/loss_state.png">https://mywebspace.net.au/UOWC128/assignment_4/images/loss_state.png</a>
Error Image	<a href="https://mywebspace.net.au/UOWC128/assignment_4/images/error_state.png">https://mywebspace.net.au/UOWC128/assignment_4/images/error_state.png</a>
Audio File	<a href="https://mywebspace.net.au/UOWC128/assignment_4/audio/applause.mp3">https://mywebspace.net.au/UOWC128/assignment_4/audio/applause.mp3</a>

<code>input</code> Element	For Random Values	For User Input
Border Effect	1px dotted <a href="#">#ffa500</a>	1px dotted <a href="#">#0080ff</a>
Background Colour	<a href="#">#fff9ef</a>	<a href="#">#ecf2f4</a>
Foreground Colour	<a href="#">#ffa500</a>	<a href="#">#0080ff</a>

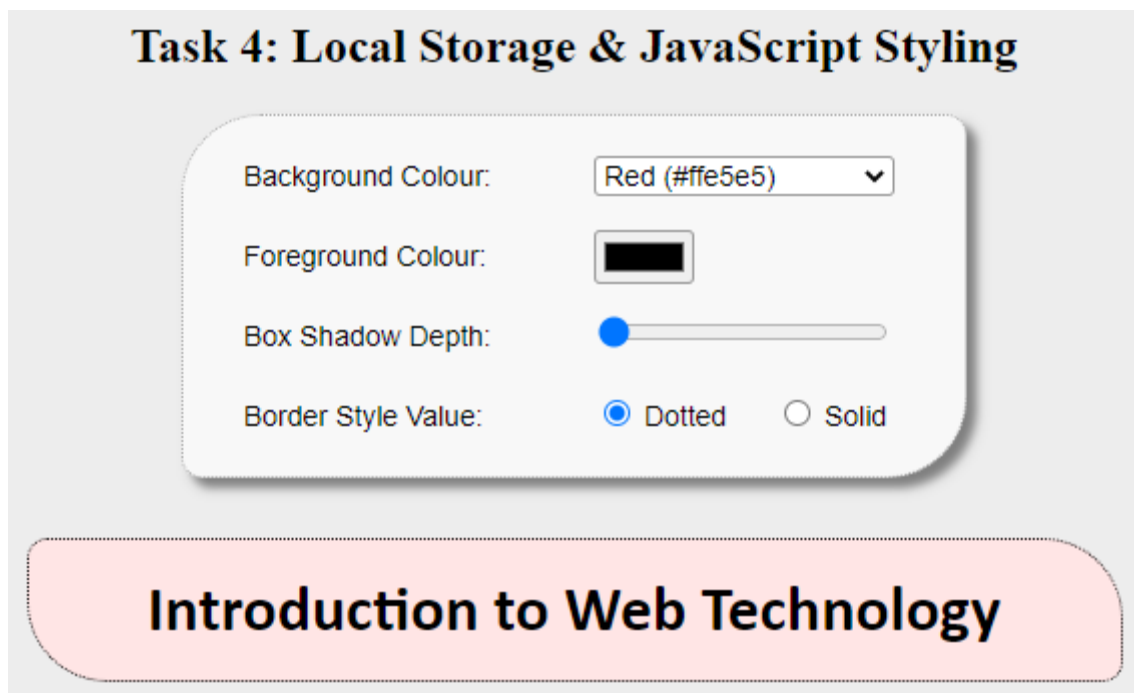
Border Effect (button)	1px dotted <a href="#">#008000</a>
Background Colour (button)	<a href="#">#ecf4f2</a>
Foreground Colour (button)	<a href="#">#008000</a>

**Please Note:**

- Your solution must use *more than one* user-defined JavaScript function.
- HTML tables should **not** be used to layout this application interface.
- Save your completed file as “**A4\_TASK3.html**”.
- This will be the third of 6 *separate files* that you must submit for this assignment.

**This task is worth 2 marks.**

**Task 4:** Create the HTML and CSS code needed to reproduce the following web application interface. Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown below.



**Design Requirements:**

- The heading text should be displayed within an `h2` element.
- The container for the `select` and `input` elements must:
  1. Show a matching border and background colour (see “**Design Resources**”)
  2. Contain a border radius effect with a shorthand value of `40px 10px`
  3. Apply an `arial-14px` font displayed with a `left` alignment
  4. Be set to a width of `360px` and contain a matching shadow effect (see “**Design Resources**”)
  5. Include padding to reflect the proportions shown in the above illustration
- The `select` element within the container should:
  1. List the six specified colour names (see [Figure 1](#)) and their corresponding hexadecimal values
  2. Utilise option groups to categorise “Pastel” colours and “Primary” colours
  3. Be set to a width of `150px` and inherit both the font family and font size of its parent element
- The `input` elements within the container must:
  1. Contain the correct `type` attribute values to match the above illustration
  2. Be grouped where appropriate to ensure proper functionality

- Default values should be specified for all `select` and `input` elements (see “**Design Resources**”).
- A `label` element must be used for each text description included within the container.
- The “Introduction to Web Technology” text should:
  1. Be declared as an `h1` element with a width of `525px` and a height of `50px`
  2. Contain a border radius effect with a shorthand value of `10px 40px`
  3. Apply a `calibri` font along with a matching horizontal and vertical text alignment
  4. Include padding to reflect the proportions shown in the above illustration
  5. Display styling that matches the values set in each `select` and `input` element
- The following actions must occur when the HTML document is loaded:
  1. All values recorded in `localStorage` are retrieved
  2. All elements within the container are set to the retrieved values
  3. All retrieved values are applied to the “Introduction to Web Technology” text

**Note: Apply the default for any style value that is not available from local storage**
- The following actions should occur when any `select` or `input` values are changed:
  1. The updated values are recorded in `localStorage`
  2. The updated values are applied to the “Introduction to Web Technology” text (see [Figure 2](#))

**Design Resources:**

Border Effect (container)	1px dotted <a href="#">#a8a8a8</a>
Background Colour (container)	<a href="#">#f8f8f8</a>
Shadow Effect (container)	5px 5px 5px <a href="#">#808080</a>

Default Styling Values			
Background Colour	<a href="#">#ffe5e5</a>	Box Shadow Depth <sup>1</sup>	0
Foreground Colour	<a href="#">#000000</a>	Border Style Value	dotted

<sup>1</sup> The “**Box Shadow Depth**” range should allow a minimum value of 0 and a maximum value of 7.

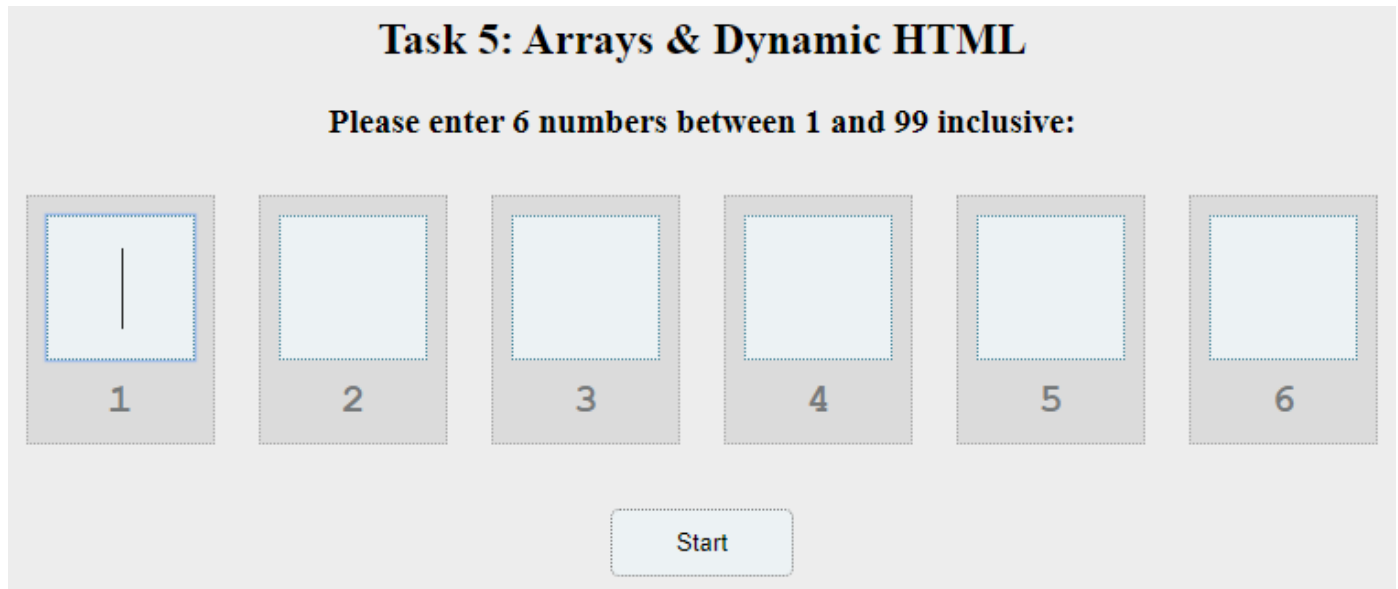
**Please Note:**

- Your solution must use *more than one* user-defined JavaScript function.
- HTML tables should **not** be used to layout this application interface.
- Save your completed file as “**A4\_TASK4.html**”.
- This will be the fourth of *6 separate files* that you must submit for this assignment.

**This task is worth 2 marks.**



**Task 5:** Create the HTML and CSS code needed to reproduce the following web application interface. Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown below.



The image shows a web application interface for Task 5. At the top, there is a heading "Task 5: Arrays & Dynamic HTML" in a bold, serif font. Below the heading is a sub-heading "Please enter 6 numbers between 1 and 99 inclusive:" in a bold, serif font. The main part of the interface consists of six input fields, each with a label below it. The labels are "1", "2", "3", "4", "5", and "6", displayed in a bold, serif font. Each input field is a light blue square with a dashed border. The first input field contains a vertical line cursor. Below the input fields is a "Start" button, which is a light blue rectangle with a dashed border and the word "Start" in a bold, serif font.

**Design Requirements:**

- The heading text should be displayed within an `h2` element.
- The sub-heading text must be displayed within an `h3` element.
- The containers for each `label` and non-button `input` element should:
  1. Be declared as `div` elements with a CSS `display` value of `inline-block`
  2. Show a matching border and background colour (see “**Design Resources**”)
  3. Contain a margin and padding value set to `10px`
- The `label` element within each container must:
  1. Be displayed with a bold `22px` `courier` font styling
  2. Show a matching foreground colour (see “**Design Resources**”)
  3. Include suitable padding to reflect the proportions shown in the above illustration
- The `input` element within each container should:
  1. Be declared as a `type="text"` variant with a height and width of `75px`
  2. Contain an `arial-40px` font and bold styling
  3. Show a matching border, colouration (see “**Design Resources**”) and alignment
  4. Contain a `maxlength` attribute to limit user input to a maximum of two (2) characters
- An `autofocus` attribute must also be used to pass focus to the first `input` element.
- A button labelled “Start” should be included and styled as follows:
  1. Contains a width of `100px` and a padding of `10px`
  2. Shows a matching background colour (see “**Design Resources**”)
- The following actions must occur when the “Start” button is clicked:
  1. The user’s input is validated

**If the input is valid** (see [Figure 1](#)):

  2. The user’s input is inserted into an array and sorted appropriately
  3. A summary table is generated and displayed within a predefined `div` element

**If the input is invalid** (see [Figure 2](#)):



2. Appropriate feedback is presented to the user within a predefined `div` element
3. No subsequent actions are taken

- The summary table should have a matching border and colouration (see “**Design Resources**”).
- The first column of the table must be composed of `th` elements.
- All average and median values should be rounded to two (2) decimal places.
- A “pointer” CSS cursor effect must be applied to all `label` elements and interface buttons.
- The above-mentioned effect should only be rendered when the mouse hovers over these components.

**Design Resources:**

Border Effect (container)	1px dotted <a href="#">#a8a8a8</a>
Background Colour (container)	<a href="#">#dbdbdb</a>
Foreground Colour (label)	<a href="#">#76797a</a>
Border Effect (input)	1px dotted <a href="#">#468499</a>
Background Colour (input)	<a href="#">#ecf2f4</a>
Foreground Colour (input)	<a href="#">#2f3030</a>
Background Colour (button)	<a href="#">#ecf2f4</a>

‘Quantitative Summary’ Table			
Border Effect	1px dotted <a href="#">#468499</a>	Foreground Colour <sup>(th)</sup>	<a href="#">#525455</a>
Background Colour <sup>(th)</sup>	<a href="#">#ecf2f4</a>	Foreground Colour <sup>(td)</sup>	<a href="#">#2f3030</a>

**Please Note:**

- You must use the same “[Error Image](#)” provided in **Task 3** to present validation feedback to the user.
- Save your completed file as “**A4\_TASK5.html**”.
- This will be the fifth of 6 separate files that you must submit for this assignment.

**This task is worth 2 marks.**

**Task 6:** Create the HTML and CSS code needed to reproduce [this web application interface](#) (see [Figure 1](#)). Extend your code by including the JavaScript functionality listed in “**Design Requirements**”. You must use the same layout and inclusions as shown within the interface imagery.

**Design Requirements:**

- The heading text should be displayed within an `h2` element.
- The sub-heading text must be displayed within an `h3` element.
- A button labelled “Start” should be included and styled as follows:

1. Contains a width of 100px and a padding of 10px
  2. Shows a matching background colour and CSS cursor effect (see “**Design Resources**”)
- The following actions must occur when the “Start” button is clicked:
    1. A total of six (6) JavaScript objects should be created and stored within an array
      - The first object must contain the headings for each table column (see [Figure 2](#))
      - The remaining objects should contain the table records (one record per object)
    2. The entire ‘Subject’ table (see [Figure 2](#)) must be recreated using [JavaScript DOM Nodes](#)
    3. A suitable variable must be used to sum the credit points for each subject
    4. The completed table should be appended to a predefined div element
    5. The credit point total should then be appended to the same predefined div element
  - The div element should be styled with a calibri-18px font.
  - The ‘Subject’ table must use th and td elements where appropriate.
  - The table should show a matching border, alignment and background colour (see “**Design Resources**”).
  - :nth-child() must be used to shade even and odd table rows (see “**Design Resources**”).
  - The credit point total should be displayed within an h4 element.

**Design Resources:**

Background Colour (button)	<a href="#">#ecf2f4</a>
Cursor Effect (button)	pointer

‘Subject’ Table			
Border Effect	1px solid <a href="#">#91adb7</a>	:nth-child(even)	<a href="#">#f5f8f9</a>
Background Colour (th)	<a href="#">#abc7d1</a>	:nth-child(odd)	<a href="#">#e3ecef</a>

**Please Note:**

- Save your completed file as “A4\_TASK6.html”.
- This will complete the set of 6 separate files that you must submit for this assignment.

**This task is worth 2 marks.**

**The following applies to all relevant components of your submission:**

Additional Marking Criteria	Available Marks
All documents validate <u>without issue(s)</u> using the W3C validation services	1.0
All scripts function <u>without issue(s)</u> in Google Chrome Developer Tools	1.0
All code is neat and suitably indented (2 Spaces Only)	0.5

Documents contain explanatory commenting where appropriate	<b>0.5</b>
--	------------

**- THIS IS THE END OF THE ASSIGNMENT -**

Version 1.0 by Jason Rodriguez