HW 1 – Client-Side

Contents

[0 Overview 1](#_Toc158203316)

[1 Problem 1 – HTML (2 hrs) 2](#_Toc158203317)

[2 Problem 2 – CSS (2 hrs) 2](#_Toc158203318)

[3 Problem 3 – Java Script (4 hrs) 4](#_Toc158203319)

[4 Problem 4 – jQuery (3 hrs) 5](#_Toc158203320)

[5 Problem 5 – XML (0.5 hrs) 5](#_Toc158203321)

[6 Problem 6 – JSON (3 hrs) 6](#_Toc158203322)

[7 Problem 7– Responsive Design (4 hrs) 7](#_Toc158203323)

[8 Problem 8 – Bootstrap (4 hrs) 8](#_Toc158203324)

[9 Submission 8](#_Toc158203325)

# Overview

There are 8 problems in this assignment and the estimated development time is about 23 hours. It may take a little less, or much more, depending on your skill. In total, this assignment is 19% of your final grade. The breakdown for each problem is shown below:

|  |  |  |
| --- | --- | --- |
| **HW 1** | **Estimated Development Time (hrs)** | **Weight** |
| Problem 1 | 2 | 10% |
| Problem 2 | 2 | 10% |
| Problem 3 | 4 | 20% |
| Problem 4 | 3 | 15% |
| Problem 5 | 0.5 | 2% |
| Problem 6 | 3 | 15% |
| Problem 7 | 4 | 10% |
| Problem 8 | 4 | 20% |
| **Total** | **22.5** | **100%** |

# Problem 1 – HTML (2 hrs)

Create a VS project with the name: *hw1\_lastName*.

Write these web pages:

1. *default.html* –
	1. Has a header with your name.
	2. Below the header there is a row with 8 links: Problem 1, Problem 2, …, Problem 8. The “Problem 1” link links to a page named: *about.html.* The other links link to the subsequent problems in this HW and will be describe as you go along. Be sure to come back to this and supply the links as you work on subsequent problems.
2. *about.html* – Write this page telling me about yourself (anything at all, for example, any/all/ or none of these: interests, hobbies, computing interests, career hopes, *etc*.
	1. You must use all of these tags: title, p, table (with headers, td, and using rowspan or colspan), a, ul, ol. You can use other tags.
	2. You must have a link to a page, *page2.html* in a subfolder named: *info. page2.html* simply has a link titled *back* which links back to *about.html.*
	3. Provide a link, “Home” that returns to *default.html.*

# Problem 2 – CSS (2 hrs)

Write a web page named: *css.html*.

1. Modify the “Problem 2” link on *default.html* to link to this page (*css.html*).
2. Provide 5 examples of using CSS.
	1. You must use either internal styles or an external style sheet but NOT inline styles.
	2. The first example illustrates an element selector; the second example illustrated a class selector; third uses an id selector; fourth uses an anonymous selector; and fifth uses a descendent selector.
	3. Each example must show actual CSS followed by the example. See example below.
	4. Each example is separated by a horizontal rule. See example below.
3. You are permitted to do additional examples and/or have multiple (linked) pages.

Example:



To display code, you should surround it in <pre><code> tags as shown below. Code for example above:

<h3>Element Selector-CSS Code</h3>

<pre><code>

 div{

 margin-left:auto;

 margin-right:auto;

 padding:20px;

 border:10px solid #0066FF;

 background-color:#00FFFF;

 margin-top: 30px;

 letter-spacing: 2px;

 font-weight:bold;

 width:75%;

 }

</code></pre>

<h3>Element Selector-Example</h3>

<div>

 “Who wants to die? Everything struggles to live. Look at that tree growing up there out of that grating. It gets no sun, and water only when it rains. It's growing out of sour earth. And it's strong because its hard struggle to live is making it strong. My children will be strong that way.”

 ― Betty Smith, A Tree Grows in Brooklyn

</div>

<hr style="border-top: 3px solid black;" />

<h3>Class Selector-CSS Code</h3>

# Problem 3 – Java Script (4 hrs)

Write a page named: *javascript.html* that appears similar to what is shown on the right. The page should fulfill these requirements:

1. Modify the “Problem 3” link on *default.html* to link to this page.
2. Ignore the bread crumbs: “HW 1 > JavaScript Example”. You do not need this on your page.
3. The easiest approach is to use a table to organize the input portion of the page. 2 columns are all that is needed. You are allowed to use a more advanced approach.
4. All code must be written in JavaScript (not jQuery).
5. The multi-select box must display these exact 5 books with the corresponding prices:
	1. Walden Pond ($20)
	2. Zen and the Art…($15)
	3. The Razor’s Edge ($8)
	4. The Poisonwood Bible ($15)
	5. Wuthering Heights ($10)
6. The user can select multiple books from the drop down.
7. The user can optionally supply a book with a title and price in the textboxes.
8. A rush order is $10 and gift wrapping is $5.
9. A general customer receives no discount and is the default. A government customer receives a 10% discount on the entire purchase.
10. The dropdown lists these exact three states and the corresponding price to ship to that state:
	1. NY ($5)
	2. CA ($7)
	3. SC ($3)
11. When the purchase button is pressed the complete order is neatly displayed along with the total price. For example:

|  |  |  |
| --- | --- | --- |
| Example 1 |  | Example 2 |
| Order----------------------$15.00 – Zen and the Art…$15.00 – The Poisonwood BibleGeneral Customer$10.00 – Rush order$3.00 – Ship to SC----------------------Total: $43.00 |  | Order----------------------$15.00 – Zen and the Art…$15.00 – The Poisonwood Bible$25.00 – On the Road (user specified)Government Customer$10.00 – Rush order$5.00 – Gift wrap$3.00 – Ship to SC----------------------Sub Total: $73.00Discount (10%): $7.30Total: $65.70 |

1. Each time the user presses the purchase button, a fresh calculation is done based on whatever is supplied, selected, *etc.* In other words, if the user supplies the data in Example 1 and presses Purchase, then the result is as shown above. Then, if they change from “General Customer” to “Government Customer”, and remove a book, then the correct total should be displayed. **Your code will doe this correctly as long as you do not use any global variables.**

# Problem 4 – jQuery (3 hrs)

For this problem, you will redo Problem 3 with jQuery.

1. Write a page named: *jQuery.html*. You could consider copying: *javascript.html* and stripping out all the Java Script.
2. Modify the “Problem 4” link on *default.html* to link to this page.
3. Implement all functionality and data access with *jQuery*.

# Problem 5 – XML (0.5 hrs)

For this problem:

1. Create a well-formed XML file for something that interests you and name this file, *myxml.xml*. The root should have at least 3 child elements.
2. Modify the “Problem 5” link on *default.html* to link to this page.

# Problem 6 – JSON (3 hrs)

For this problem:

1. Create a page, *json.html* that reads [jSON](https://raw.githubusercontent.com/drgap/json_example/main/company.json) from GitHub and displays it as shown below. The display should be similar to the one below.

Note: Your code must use the link to the raw page: <https://raw.githubusercontent.com/drgap/json_example/main/company.json>

1. Modify the “Problem 6” link on *default.html* to link to this page.



# Problem 7– Responsive Design (4 hrs)

For this problem:

1. Create a page, *responsive.html*.
2. Modify the “Problem 7” link on *default.html* to link to this page.
3. On *responsive.html*, build a responsive page based on the wire frame below, which is similar to a Stack Overflow [page](https://stackoverflow.com/questions/3971049/whats-the-c-version-of-javas-arraylist).

Requirements:

* Must use only CSS. No Bootstrap, nor other framework.
* Must use CSS Flexbox and/or Grid.
* The structure of the page should be as close as possible to what is shown in the wireframe.
* The content of the page should be realistic and cohesive. You can copy from the Stack Overflow page, or you can use some other context. By “content”, we mean everything on the page: text, link titles, headings, titles, etc. You can model exactly what is shown: 2 categories on the left, 2 items on the right, a question and 2 answers.
* The links should work and they can simply link to the page itself.
* The structure and content are styled with CSS in any way you want.
* The page should be responsive at at-least 2 screen sizes.

Rubric:

|  |  |
| --- | --- |
| **Criterion** | **Weight** |
| Structure consistent | 30% |
| Structure style | 15% |
| Content cohesive | 30% |
| Content style | 15% |
| Responsiveness | 10% |

Wireframe:



# Problem 8 – Bootstrap (4 hrs)

For this problem:

1. Create a page named: *bootstrap.html* and modify the “Problem 8” link on *default.html* to link to this page.
2. Build a Bootstrap page about something that interests you, that meets these requirements:

|  |  |  |
| --- | --- | --- |
| **Num** | **Met?** | **Requirement** |
|  |  | The contents of the page has a common theme. It is not just random stuff |
|  |  | The content is original unless cited. Any cited content must is used sparingly and be appropriately used. |
|  |  | The page is responsive to at least three different viewport sizes. |
|  |  | The page has a header, navbar, content, and footer. Links are required, at least in the navbar, but do not have to link anywhere. |
|  |  | The page uses at least 1 responsive image |
|  |  | The page uses at least 1 of these: Table, List/List Group, Button/ButtonGroup |
|  |  | The page uses at least 1 of these: Cards, Accordion, Tabs, Carousel, Scroll Spy |

You can use either of these templates: [w3schools](https://www.w3schools.com/bootstrap5/tryit.asp?filename=trybs_template1), [TutorialRepublic](https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=responsive-layout) as a starting point, or any other one.

# Submission

Zip all (and only) the required files for all 8 problems, and including the *default* page into a file named: hw1\_lastname.zip. Submit on Blazeview in the HW 1 dropbox by the due date.

1. Do the following:
2. Select your project folder (*hw1\_lastName*) in File Explorer
3. Right-click and choose: *Send to, Compressed (zipped) folder.* It should be named: *lab01\_lastName.zip* (which probably is the default name).
4. **Before submitting: copy the zip file to some other location, unzip and launch the solution file (*lab01\_lastName.sln*) and make sure everything works.**
5. Submit the zip file on blazeview by the due date.