Name: _____________________________________________

TA / Section: _____________________________________________

Student ID #: _____________________

Rules:
- You have **60 minutes** to complete this part of the exam.
  You may receive a deduction if you keep working after the instructor calls for papers.
- This test is open-book/notes. You may use any paper resources other than practice exams.
- You may *not* use any computing devices, including calculators, cell phones, or music players.
- Unless otherwise indicated, your code will be graded on proper behavior/output, not on style.
- Please do not abbreviate code, such as writing ditto marks (""") or ellipses (...).
- If you enter the room, you must turn in an exam and will not be permitted to leave without doing so.
- You must show your **Student ID** to a TA or instructor for your submitted exam to be accepted.

Good luck!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Earned</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HTML/CSS Tracing</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>HTML/CSS Coding</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>JavaScript/DOM</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>TOTAL</td>
<td>Day's Total Points</td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Earned</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>?</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>TOTAL</td>
<td>Day's Total Points</td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

**TOTAL** | Exam Total Points | **100** |
1. **HTML/CSS Tracing**

   Draw a picture of how the following HTML and CSS code will look when the browser renders it on-screen. Assume that the HTML is wrapped in a valid full page with a `head` and `body`. Indicate a background coloring by shading lightly or by drawing repeated diagonal lines like this. If you can't clearly write *italic* text, *underline* it instead.

   **HTML:**

   ```html
   <p class="pliny elder">
     I <br/>don't know about you
   </p>

   <ul id="pliny">
     <li>foo</li>
     <li><em>bar</em></li>
     <li>baz</li>
   </ul>

   <h1 id="firestone">Happy day!</h1>

   **CSS:**

   ```css
   h1, p, ul { margin: 0; }
   .pliny { float: right; margin: 25px; }
   #firestone, #pliny { padding-left: 1em; }
   .elder, h1 { font-size: 200%; }
   li { display: inline; }
   .elder { width: 200px; border: solid 2px black; }
   #pliny { background-color: lightgray; border-bottom: dashed 2px black; height: 100px; }
   #ninkasi, .pliny.younger { display: block; }
   ```

   box.png is a 100×100px image of a box with an ‘X’ through it:
2. HTML/CSS Coding

Write the XHTML and CSS code necessary to recreate the following appearance on-screen.
(Adapted from the W.A.B.L page of the Washington Beer Commission website, washingtonbeer.com.)

Most of the XHTML code is given to you; the code given may not be modified. The only change you may make to the provided XHTML code is that you may add any number of `div` and/or `span` elements, possibly with `id` and/or `class` attributes, as targets for CSS styling. Write valid code that would pass the W3C validators. Assume that the given XHTML code would appear inside the `body` of the page.

- Text on the page uses a sans-serif font. The area around the outside of the page has a yellowish color of `#FFB131`.
- The overall central page area has a white background, and a 5px-thick solid black border on its bottom edge only. It is centered within the page and is 950px wide.
- The list of links appears without bullets, displayed all in one line, centered on the page. Each link's text is bold, not underlined, colored `#FFB131`, has a line height of 2em, and has 1.5em of horizontal spacing on either side of it. (Between two adjacent links there is a total of 3em space.)
- The main content area (beneath the navigation links) has 1.5em of horizontal space on the left and right. All the headers inside of it are underlined.
- The “Join WA.B.L” section is located on the right side of the main content area, with text flowing around it. Its font size is 190% of the font size of the rest of the page, and its background is colored `#FBC57B`. There is .75em of space separating its content from its edges, as well as 1em of space separating its edges from the surrounding content.

All other appearance is unspecified, and is subject to the default rendering of the browser.

Mark up the text on the next page with your `div`/`span` tags. If a tag can't fit in the space provided, write it in the margins and draw an arrow to where it should be inserted.

*Write your answer on the next page.*
2. **HTML/CSS Coding (writing space)**

Mark up the **HTML code** below.

```html
<h1><img src="header.gif" alt="Washington Beer - Home of the WASHINGTON BEER Commission" /></h1>

<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Festivals</a></li>
<li><a href="#">Breweries</a></li>
<li><a href="#">News</a></li>
<li><a href="#">WABL</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Contact</a></li>
</ul>

<h2>What is WA.B.L.?</h2>
<p>WA.B.L. is a community of Washington Beer ...</p>
<p>It's a Washington Beer Fan Club!</p>

<a href="#">JOIN WA.B.L NOW!</a>

<h2>WA.B.L. Perks</h2>
<p>WA.B.L.'ers enjoy monthly invitation-only events at various ...</p>
<p>Each year, upon renewal of their membership, WA.B.L.'ers receive a cool t-shirt ...</p>

<h2>WA.B.L. Passport</h2>
<p>WA.B.L.'ers are also encouraged to visit breweries when they're out traveling ...</p>
<p>Collecting stamps from WA breweries can get you some great prizes!</p>

Write your **CSS code** here. Put your CSS in two columns if you need more writing space, and/or use scratch paper.
3. JavaScript/DOM

**Background:** The first step in brewing beer is to extract sugars from grains by soaking the grains in hot water. This step is called *mashing*, and the resulting sugar-water is called *wort*. Each variety of grain contributes a different amount of sugar (the grain’s *yield*) to the wort. Calculating the final amount of sugar in the wort is an important part of designing a beer recipe, since the sugars will later ferment into alcohol. To measure the amount of sugar in the wort we use *specific gravity* (SG), a decimal number (e.g., 1.054) which represents the ratio of the sugar-water’s density to that of pure water (=1.000 SG). Specific gravity measurements during the brewing process are typically between about 1.010 and 1.100.

Your job is to write the **JavaScript code** to add behavior to the following page, which estimates the specific gravity of a list of user-entered grains:

![Mash-It!](image)

**HTML:**

```html
<h1>Mash-It!</h1>

<fieldset>
  <legend>Malt bill:</legend>
  <table id="malt_bill">
    <tr><th>lbs</th><th>Grain</th><th>PPG</th></tr>
    <tr><td>8.5</td><td>American 2-Row Pale</td><td>38</td></tr>
    <tr><td>.5</td><td>Cara-Pils/Dextrose</td><td>32</td></tr>
    <tr><td>.5</td><td>Crystal 20</td><td>34</td></tr>
    <tr><td>.5</td><td>Crystal 40</td><td>34</td></tr>
  </table>
</fieldset>

<button id="add_grain">Add another grain</button>
<button id="calculate">Calculate SG</button>

Estimated Specific Gravity: **1.068**
(assumes a 5.5 gallon batch)

The page consists of a table in which the user can enter grains. Every time the **Add another grain** button is clicked, a new row should be added at the bottom of the table. Each row consists of three text-entry fields, each in its own table cell, for entering the following values: an **amount** (in pounds), a **name**, and a sugar **yield** (in “PPG”). These entry fields should have the classes amount, name, and yield respectively.

When the **Calculate SG** button is pressed, your code should estimate the specific gravity of the user-entered grains by performing the following calculation:

\[
SG_{ext} = \text{round}\left(\frac{a_1y_1 + a_2y_2 + \cdots + a_ny_n}{5.5}\right) / 1000 + 1
\]

(Where \(a_i\) and \(y_i\) represent the amount and yield of the first grain, and so on.)

In other words, for each grain in the list, multiply its **amount** (the number in the “lbs” column) by its **yield** (the number in the “PPG” column), and take the sum of these products. Then divide that by 5.5 (which represents the volume of water it’s dissolved in, in gallons). Round the result to the nearest integer, then to express it as a number 1 followed by 10 decimal places (e.g., 1.054), divide by 1000 and add 1.

For example, using the values in the above screenshot, your calculation would be as follows:

\[
SG_{ext} = \text{round}\left(\frac{8.5 \cdot 38 + .5 \cdot 32 + .5 \cdot 34 + .5 \cdot 34}{5.5}\right) / 1000 + 1 = 1.068
\]

You should **assume all input to text boxes is valid**: that is, reasonable values will always be entered in the fields for amount and yield. You may assume the **Prototype** JavaScript library is loaded prior to your script being loaded. When injecting content into the page, you may use the **innerHTML** property for injecting **plain text only**; you may **not** use it to inject any HTML code.
These screenshots show various states of the program being used:

<table>
<thead>
<tr>
<th>Initial state</th>
<th>After the first grain has been entered</th>
<th>After Add another grain has been clicked</th>
<th>After another grain has been entered, and Calculate SG has been clicked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state</td>
<td>After the first grain has been entered</td>
<td>After Add another grain has been clicked</td>
<td>After another grain has been entered, and Calculate SG has been clicked</td>
</tr>
</tbody>
</table>

(Write your solution below.)

3. **JavaScript/DOM (writing space)**
3. JavaScript/DOM (additional writing space)