CSE 190 M, Summer 2011
Final Exam, Part 1 (LAB), version B
Friday, August 19, 2011

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!

Problem	Description	Earned	Max
1	HTML/CSS Tracing		15
2	HTML/CSS Coding		15
3	JavaScript/DOM		20
TOTAL	**Day's Total Points**		**50**

Problem	Description	Earned	Max
4	?		15
5	?		15
6	?		20
TOTAL	**Day's Total Points**		**50**

TOTAL	**Exam Total Points**		**100**

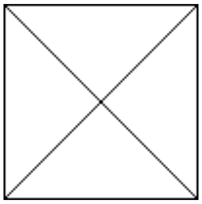
(This page intentionally left blank.)

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. Draw all sizes with **reasonably accurate proportions**. Base your proportions on the size of the image `box.png`.

HTML:

`box.png` is a 100×100px image of a box with an 'X' through it:



```
<p class="lucille">
  <span class="roger">
    <img id="lisa" src="box.png" alt="x" />
    Why     hello th<br/>ar how do
    you do good sir
    ?
  </span>
</p>

<h1 id="bob">Yee-haw!</h1>

<ul id="lucille" class="roger">
  <li>foo</li>                          <li>bar</li>
  <li class="manny"><em>baz</em></li>  <li class="manny">qux</li>
</ul>
```

CSS:

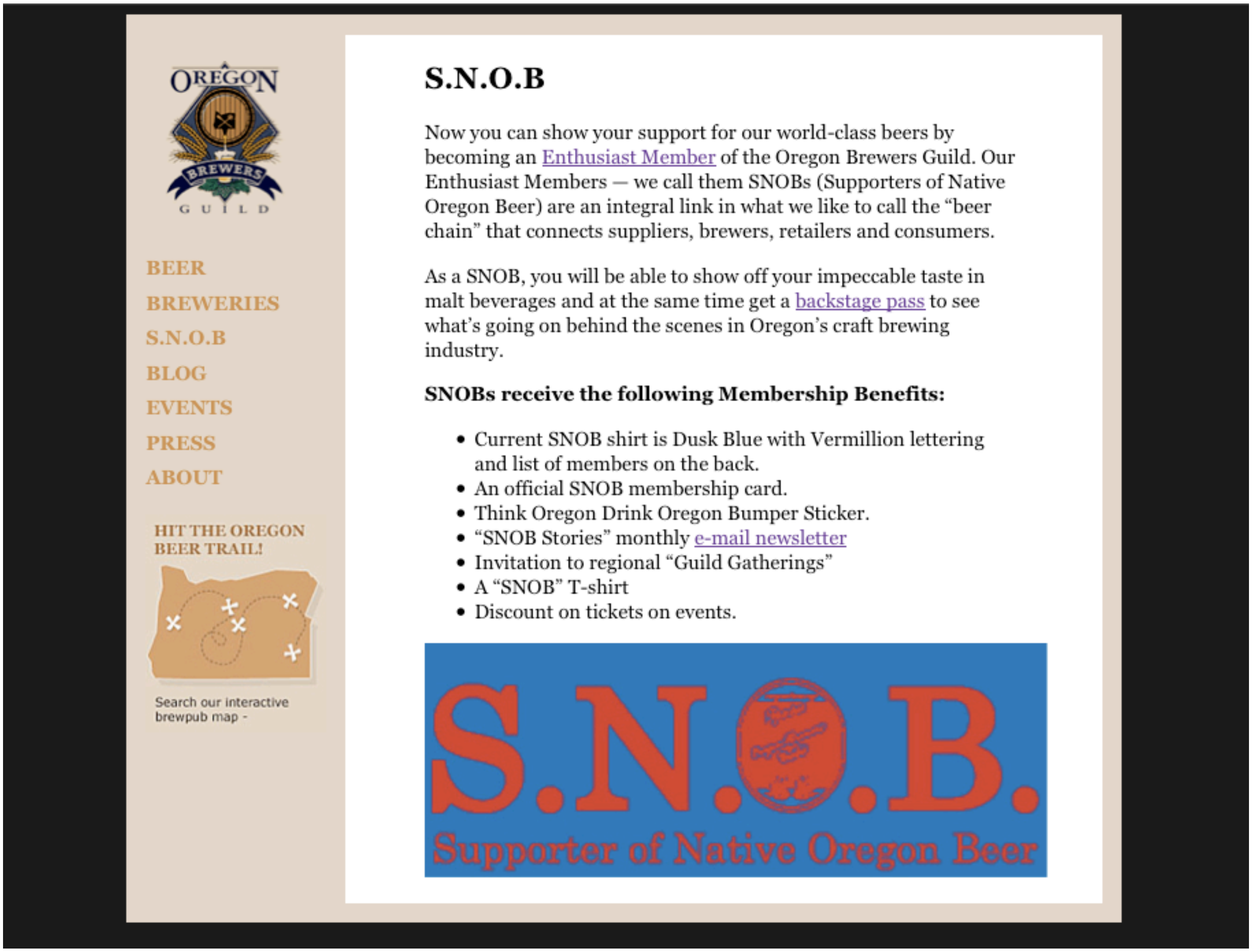
```
body { font-family: sans-serif; }
/* clear all margins and paddings initially */
body, h1, p, ul { margin: 0; padding: 0; }
```

```
#lisa    { position: absolute;
           top: 100px; right: 100px; }
.lucille { background-color: lightgray;
           border-bottom: dashed 2px black; }
.roger, .lucille { height: 100px; }
#bob, .roger     { text-align: right; }
#bob, .lucille   { font-size: 50px; }
.manny           { display: inline; }
```

```
.roger   { display: block;
           float: left;
           width: 300px;
           margin: 50px;
           border: solid 2px black; }
#lucille { clear: left;
           text-align: center; }
```

2. HTML/CSS Coding

Write the XHTML and CSS code necessary to recreate the following appearance on-screen.
(Adapted from the S.N.O.B page of the Oregon Brewers' Guild website, oregonbeer.org.)


OREGON
BREWERS
GUILD
BEER
BREWERIES
S.N.O.B
BLOG
EVENTS
PRESS
ABOUT
HIT THE OREGON BEER TRAIL!
Search our interactive brewpub map -
S.N.O.B.
Supporter of Native Oregon Beer

S.N.O.B

Now you can show your support for our world-class beers by becoming an Enthusiast Member of the Oregon Brewers Guild. Our Enthusiast Members — we call them SNOBs (Supporters of Native Oregon Beer) are an integral link in what we like to call the "beer chain" that connects suppliers, brewers, retailers and consumers.

As a SNOB, you will be able to show off your impeccable taste in malt beverages and at the same time get a backstage pass to see what's going on behind the scenes in Oregon's craft brewing industry.

SNOBs receive the following Membership Benefits:

- Current SNOB shirt is Dusk Blue with Vermillion lettering and list of members on the back.
- An official SNOB membership card.
- Think Oregon Drink Oregon Bumper Sticker.
- "SNOB Stories" monthly e-mail newsletter
- Invitation to regional "Guild Gatherings"
- A "SNOB" T-shirt
- Discount on tickets on events.

Most of the XHTML code is given to you; the code provided 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 HTML text would appear inside the `body` of the page.

- Text on the page uses the **"Georgia"** font. The area around the outside of the page has a nearly-black color of **#E2D1BB**.
- The overall page area has a background colored **#E2D1BB**. It is **centered** within the page and is **50em** wide.
- The left sidebar has a header in it, which should be **centered** within the column. The list of links in the left column are **bolded**, have **no bullets or underlines**, have a line height of **1.75em**, and are colored **#C5833B**. The sidebar's content is **8em** wide.
- The central content area has a **white** background color. Its content is **30em** wide. Additionally, there is **4em** of horizontal space on either side separating the content from the edges of the area.
- Surrounding each of the columns is **1em** of space on all sides.

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

(Write your solution below.)

2. HTML/CSS Coding (writing space)

Mark up the **HTML code** below.

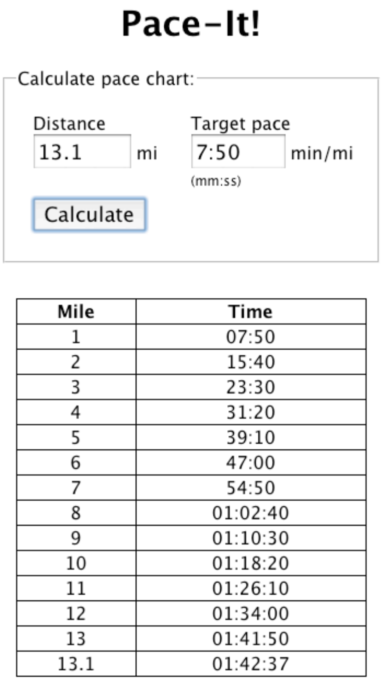
```
<h1><img src="logo.gif" alt="Oregon Brewers Guild" /></h1>


<ul><li><a href="">BEER</a></li>            <li><a href="">BREWERIES</a></li>
    <li><a href="">S.N.O.B</a></li>         <li><a href="">BLOG</a></li>
    <li><a href="">EVENTS</a></li>          <li><a href="">PRESS</a></li>
    <li><a href="">ABOUT</a></li></ul>


<p><img src="map.jpg" alt="Hit the Oregon beer trail! Search our interactive pub map." /></p>


<h2>S.N.O.B</h2>


<p>Now you can show your support for our world-class beers by becoming an <a href="">Enthusiast ...</p>
<p>As a SNOB, you will be able to show off your impeccable taste in malt beverages and at the same ...</p>


<p><strong>SNOBs receive the following Membership Benefits:</strong></p>
<ul>
  <li>Current SNOB shirt is Dusk Blue with Vermillion lettering and list of members on the back.</li>
  <li>Think Oregon Drink Oregon Bumper Sticker.</li>
  <li>“SNOB Stories” monthly <a href="">e-mail newsletter</a></li>
  ...
</ul>


<p><img src="snob.jpg" alt="2010 SNOB logo" /></p>
```

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

3. JavaScript/DOM

Background: Distance runners need to pace themselves to avoid burning out before the end of a long race. It's sometimes helpful to print out and carry a card which lists exactly what time the runner should reach each mile marker if they are running at their *target pace*. That way, if the runner reaches a mile marker at a time substantially off from the one listed, they know they are going too fast (or too slow) and need to adjust their pace accordingly.

Write the **JavaScript code** to add behavior to the following page, which creates a runner's pace chart based on entered values for distance and target pace:



Pace-It!

Mile	Time
1	07:50
2	15:40
3	23:30
4	31:20
5	39:10
6	47:00
7	54:50
8	01:02:40
9	01:10:30
10	01:18:20
11	01:26:10
12	01:34:00
13	01:41:50
13.1	01:42:37

HTML:

```
<h1>Pace-It!</h1>

<fieldset>
  <legend>Calculate pace chart:</legend>

  <div class="distance">
    <label>Distance</label><br/>
    <input type="text" id="distance" /> mi
  </div>

  <div class="target">
    <label>Target pace</label><br/>
    <input type="text" id="target_pace" /> min/mi<br/>
    <span>(mm:ss)</span>
  </div>

  <div><button id="calculate">Calculate</button></div>
</fieldset>

<table id="pace_chart">
  <tr id="header_row"><th>Mile</th><th>Time</th></tr>
  <!-- calculated pace informaiton will go here -->
</table>
```

The page consists of fields for entering a **distance** (in miles) and **target pace** (in minutes per mile), as well as a table for the **pace chart**. Initially the table contains only a header row with the column names, and no data; as such, you should **hide** the table as soon as the page is loaded. Then, when the user clicks the **Calculate** button, you should first **populate** the table with data, and then **show** it.

When the chart has been generated, each row should consist of a **mile** number and a **time** at which the runner should reach that mile, each in its own table cell. Because doing arithmetic on times is very difficult, your code should make use of the following **provided function** that will multiply times for you:

```
// returns a 'mm:ss' or 'hh:mm:ss' time string representing
// the given time 't' multiplied by the given factor 'f'
function multiplyTime(t, f) {
  ...
}
```

Assume this function has been previously declared, and you are free to make use of it in your code. You **do not** need to write/implement this function, only call it.

The last row in the table should always be for the **exact value** entered in the distance field, even if the distance is a non-integer number like 26.2. If the user clicks the Calculate button after a pace chart has perviously been generated, **all previous data should be cleared** before the new data is added. When you do this, you must **always retain the header row**; it should never be removed or re-generated.

You should **assume all input to text boxes is valid**; that is, reasonable values will always be entered in the fields for distance and pace. 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.

3. JavaScript/DOM (writing space)

(This page intentionally left blank. You may use it as additional writing space if needed.)