

CSE 190 M, Spring 2011
Final Exam, Thursday, June 9, 2011

Name: _____

Quiz Section: _____ **TA:** _____

Student ID #: _____

Rules:

- You have **110 minutes** to complete this 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 dot-dot-dot marks (...).
- 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		20
2	HTML/CSS Coding		20
3	PHP		20
4	JavaScript/DOM		20
5	Regular Expressions		20
TOTAL	Total Points		100

(This side 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 non-white background color by shading lightly or by drawing diagonal lines like this. Note that it is possible that some of the rules shown will not apply to any elements.

HTML:

```
<div id="b">
  <div>a</div>    <div>b</div>
  <span>x</span>   <span>y</span>
</div>

<div class="a">
  <span class="b c">one <br /> two</span>
  <span id="d">three <br /> four</span>
</div>
```

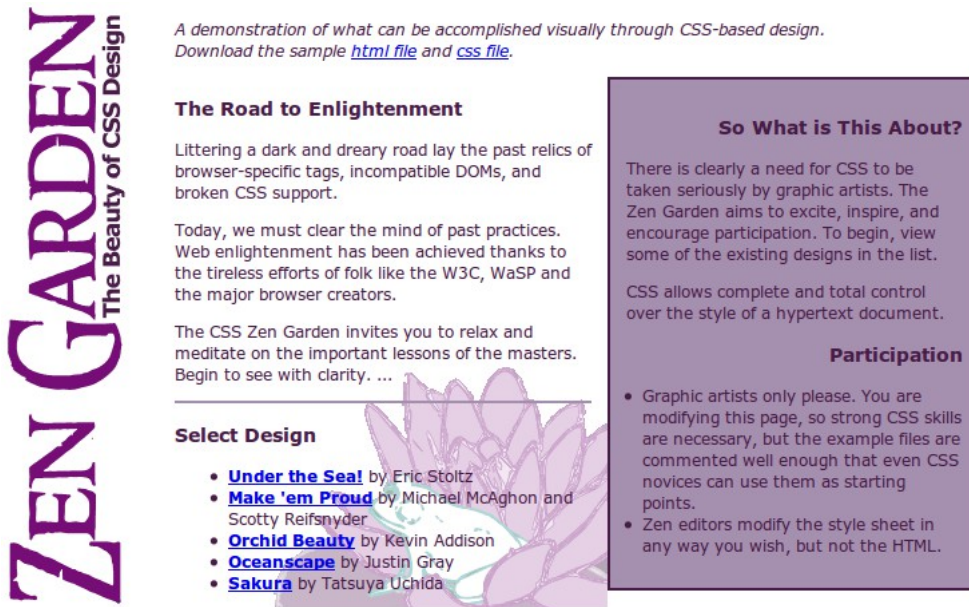
CSS:

```
.a, #b > div { margin: 2em; }
.a .d       { text-decoration: underline; }
#b          { width: 50%; border: 2px dashed black; }
div > .c, #b { background-color: cyan; }
div         { background-color: white; }
div div     { text-align: right; padding-right: 4em; border: 2px dashed black; }
```

2. HTML/CSS Coding

(Based on "Lily Pond" theme from <http://csszengarden.com/>.)

Write the HTML/CSS necessary to recreate the following appearance. No manual line breaks have been inserted.



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

- The image `header.gif` appears once at the top-left corner of the page. This image contains the vertical "Zen Garden" text seen in the image above. Other content is shifted right by 130px to avoid covering this background image. The page's usual Zen Garden header text should not be shown because the image replaces it.
- Text on the page uses a font named "Verdana", or if that font is not available, the default sans-serif font available on the system. Text on the page is displayed in the color `#471C47`. The overall default font size is 9pt.
- The "A demonstration of..." paragraph's text is in italic.
- There is a first column of content, 26em wide, that displays "The Road to Enlightenment" and "Select Design" and so on. Behind that content is the image `toad.jpg` aligned to the bottom-right corner of the area. 1em of space separates text in this area from the other column to its right. A 3px-thick solid border of color `##A690AF` separates the "Road to Enlightenment" content from the "Select a Design" content.
- The links in the "Select Design" area are in bold. (Other links on the page are not bold.)
- There is a second column of content, 21em wide, that displays "So What is This About?" and "Participation". It has a 2px-thick border of color `#471C47` and a background color of `#A690AF`. It touches the right edge of the first column previously described. 1em of space separates text in this area from its border on all sides. If the page is resized to be wider than shown above, the columns remain in the same orientation, connecting on their edge.
- Headings in the second column are right-aligned.

You may not use absolute or fixed positioning when solving this problem.

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 to insert it. Any elements not described use the default browser styles.

Write your answer on the next page.

2. HTML/CSS Coding (writing space) - Mark up the HTML code below.

```
<h1>Zen Garden</h1>

<p>A demonstration of what can be accomplished visually through CSS-based design. <br />
  Download the sample <a href="">html file</a> and <a href="">css file</a>.</p>

<h3>The Road to Enlightenment</h3>
<p>Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible ...</p>
<p>Today, we must clear the mind of past practices. Web enlightenment has been achieved ...</p>
<p>The CSS Zen Garden invites you to relax and meditate on the important lessons of the ...</p>

<h3>Select a Design</h3>

<ul><li><a href="">Under the Sea!</a> by Eric Stoltz</li>    <!-- the link hrefs don't matter -->
  <li><a href="">Make 'em Proud</a> by Michael McAghon and Scotty Reifsnyder</li>
  <li><a href="">Orchid Beauty</a> by Kevin Addison</li>
  <li><a href="">Oceanscape</a> by Justin Gray</li>
  <li><a href="">Sakura</a> by Tatsuya Uchida</li></ul>

<h3>So What is This About?</h3>
<p>There is clearly a need for CSS to be taken seriously by graphic artists. The Zen...</p>
<p>CSS allows complete and total control over the style of a hypertext document. ...</p>

<h3>Participation</h3>
<ul><li>Graphic artists only please. You are modifying this page, so strong CSS skills are...</li>
  <li>Zen editors modify the style sheet in any way you wish, but not the HTML.</li></ul>
```

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

3. PHP

Write the PHP code for a partial web page that searches for names that contain a given letter a given number of times, case-insensitively. Your web service would be located in a file named `q4.php` on the server. In this same directory is a file `peeps.txt`; each line of this file contains a name. For example:

```
Helene Martin
Robert Purple
Marty Stepp
Stuart Reges
Bob Loblaw
```

The names are guaranteed to be non-empty and unique.

Your web service accepts a query parameter named `letter` indicating the character to be searched for, and a parameter named `times` indicating how many times that character must occur, case-insensitively. For each name that contains the letter at least the given number of times, an HTML paragraph should be printed indicating this with the name in bold along with the number of times it contains the letter. For example, if the following query is given, is given, this output is produced:

```
q4.php?letter=R&times=2
```

Robert Purple contains 'R' exactly 3 times.

Stuart Reges contains 'R' exactly 2 times.

If no names contain the given character the given number of times, output a message saying so:

```
q4.php?letter=x&times=1
```

No name contained 'x' enough times.

If one or both of the required query parameters is not passed, your service must produce an HTTP 400 Invalid Request error. You should also generate an HTTP 400 error if the `letter` parameter is not a one-letter string or if the `times` parameter is not a positive number. (Any `times` value ≤ 0 is invalid and anything else is valid.)

```
q4.php?letter=thingy&times=-4
q4.php?letter=t
q4.php?times=7
q4.php
```

HTTP/1.1 400 Invalid Request

Your code should *not* output a complete HTML page. Output a partial page containing only the paragraphs indicated. Do not use any `print` or `echo` statements in your code.

Write your answer on the next page.

3. PHP (writing space)

4. JavaScript/DOM

Write the **JavaScript code** to add behavior to the following page that has a user interface for "blending" a name. The UI allows the user to type his/her name into a text box. The user can click a button to "blend" the name, which causes each character of the name to be individually injected into a `div` with the `id` of `output`. Your code should enable the user to choose a font to use for displaying the characters of the name, which is either Arial, Comic Sans MS, or Times New Roman. The letters of the name will be displayed in the font chosen.

The letters can be positioned differently by choosing one of two radio buttons. If the "Random" button is checked, each letter of the name is positioned randomly with an x-coordinate between 0-300 pixels from the left edge of the `output` area, and a y-coordinate between 0-100 pixels from the top edge of the `output` area. If the "Sequential" button is checked, letters are placed at 15-pixel intervals, with the first at (x=15, y=15) within the `output` area, and the second at (x=30, y=30), the third at (x=45, y=45), and so on. (If there are a lot of letters, they might extend outside of the `output` area, but you don't need to worry about that.)

The code should work for multiple clicks of "Blend". On each click it should clear any previously inserted letters.

You may assume that Prototype and Scriptaculous are included in the page.

HTML:

```
<h1>Name in a Blender</h1>
<div>
  <span>Your name:</span>
  <input id="yourname" type="text" size="16" />
</div>
<div>
  <span>Font:</span>
  <select id="font">
    <option>Arial</option>
    <option>Comic Sans MS</option>
    <option>Times New Roman</option>
  </select>
</div>
<div>
  <span>Position:</span>
  <input id="random" name="pos" type="radio" value="rand" checked="checked" /> Random
  <input id="sequential" name="pos" type="radio" value="seq" /> Sequential
</div>
<div>
  <span>Go:</span>
  <button id="blend">Blend my name!</button>
</div>
<div id="output"> <!-- your blended characters go here --> </div>
```

These screenshots show the initial state, and after names have been typed and "Blend my name!" has been clicked.

Name in a Blender

Your name:

Font:

Position: ☒ Random ☐ Sequential

Go:

Name in a Blender

Your name:

Font:

Position: ☒ Random ☐ Sequential

Go:

p e i t h
h h
C s r r o

Name in a Blender

Your name:

Font:

Position: ☐ Random ☒ Sequential

Go:

S t u a r t

Write your answer on the next page.

4. JavaScript/DOM (additional writing space)

5. Regular Expressions

a) Write a regular expression to match a US zipcode. Zipcodes contain 5 numbers. Sometimes there is a dash after the 5 numbers which is followed by 4 more numbers.

Valid:	Invalid:
99999	abc12345
98115	123456789
67241-1234	12345-6

b) Write a regular expression to match a price. A price always starts with a dollar sign. Any amount of numbers can come before the decimal. Two numbers should always follow the decimal.

Valid:	Invalid:
\$14.99	\$14
\$1234567.00	\$134213.89money
\$.90	\$1.1a

c) Write a regular expression to match a self closing HTML tag. A tag must start with "<" and end with ">". It must have a name and can have 0, 1 or more properties. You do not need to worry about white space inside the tag.

Valid:	Invalid:
	</>
	
	

[abc]	A single character of: a, b, or c	.	Any single character
[^abc]	Any single character except: a, b, or c	\s	Any whitespace character
[a-z]	Any single character in the range a-z	\S	Any non-whitespace character
[a-zA-Z]	Any single character in the range a-z or A-Z	\d	Any digit
^	Start of line	\D	Any non-digit
\$	End of line	\w	Any word character (letter, number, underscore)
\A	Start of string	\W	Any non-word character
\Z	End of string		
(...)	Capture everything enclosed	a+	One or more of a
(a b)	a or b	a{3}	Exactly 3 of a
a?	Zero or one of a	a{3,}	3 or more of a
a*	Zero or more of a	a{3,6}	Between 3 and 6 of a