CSE 190 M, Spring 2007
Final Exam

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! You can do it!

Problem	Description	Earned	Max
1	HTML / CSS Tracing		15
2	HTML / CSS Coding		15
3	PHP		15
4	JavaScript / DOM		10
5	JavaScript / DOM		15
6	Ajax / XML		15
7	SQL		15
X	Extra Credit		1
TOTAL	**Total Points**		**100**

1. HTML / CSS Tracing

Draw a picture of how the following HTML/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 by shading lightly or by drawing diagonal lines like  . Assume that `stickman.png` is a generic picture of a stick man. It is possible that some CSS rules shown will not apply to any elements.

HTML:

```
<h1>Scene</h1> <h1>#7</h1>

<div id="a">Elizabeth: Captain Barbosa, I am here to negotiate
the cessation of hostilities against Port Royal.</div>

<div>Barbosa: <img src="stickman.png" alt="stick man" />
There are a lot of long words in there,
Miss.<br />What is it that you want?</div>

<div class="b">
  <div class="c">Elizabeth: I want you to leave</div>
  <div class="c">and never come back.</div>
</div>

<div class="c" id="d">Barbosa: I'm disinclined to acquiesce
to your request. Means &amp;quot;no.&amp;quot;</div>
```

CSS:

```
img     { vertical-align: top; width: 10em; height: 10em; }
#a      { border: 2px solid black; float: left;
          padding-right: 3em; width: 33%; }
.a #a   { background-color: yellow; }
.b div  { display: inline; }
.b, .c  { text-decoration: underline; }
.c > .b { background-color: yellow; }
.d      { float: right; }
```

1. **HTML / CSS Tracing (writing space)**

2. HTML / CSS Coding

Write the HTML and CSS code necessary to recreate the following appearance onscreen, between but not including the thick black lines. (No manual line breaks have been inserted into the text.) Assume that the code you're writing will be placed inside the body of the page. Part of your grade comes from choosing appropriate tags to match the semantics of the content. You should also write valid code that would pass the W3C validators, and separate stylistic information from HTML.

- **Inigo:** Hello. My name is Inigo Montoya. You killed my father. Prepare to die.

- **Count Rugen:** No!
- **Inigo:** Offer me money! Power, too. Promise me that!
- **Count Rugen:** All that I have and more! Please!
- **Inigo:** Offer me everything I ask for!
- **Count Rugen:** Anything you want.
- **Inigo:** I want my father back, you son of a ....

Some details about the desired appearance:

- All text uses default sizes and fonts, except for the characters' names which are written in a sans-serif font.
- The section of dialogue is 50px from the right edge of the page and occupies 16em in width. No manual line breaks have been inserted into the text.
- The text for each character's name is bold and underlined.
- The images come from the files `inigo.jpg` and `rugen.jpg` and are drawn at their default sizes.
- Rugen's lines are colored with a background of #DDDDDD or (red=221, green=221, blue=221). 5px separate these images from any surrounding content.
- 5px of vertical space separate each of the bulleted lines.

Mark up the text on the next page with your HTML tags. If a tag can't physically be written in the space provided, write it in the margins and draw an arrow to where it should be inserted in the text. Write the CSS styles on the page after next.

2. HTML / CSS Coding (writing space)

Inigo:

Hello. My name is Inigo Montoya. You killed
my father. Prepare to die.

Count Rugen:

No!

Inigo:

Offer me money! Power, too. Promise me that!

Count Rugen:

All that I have and more! Please!

Inigo:

Offer me everything I ask for!

Count Rugen:

Anything you want.

Inigo:

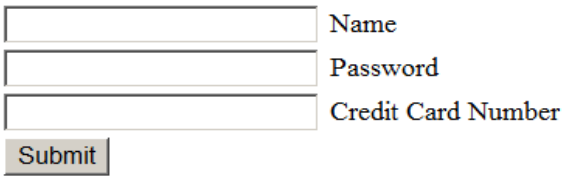
I want my father back, you son of a ....

2. HTML / CSS Coding (writing space)

3. PHP

Write PHP code that processes the following form:

```
<form action="q3.php" method="post">
    <fieldset>
        <input type="text" name="name" /> Name <br />
        <input type="text" name="pw" /> Password <br />
        <input type="text" name="cc" /> Credit Card Number <br />
        <input type="submit" />
    </fieldset>
</form>
```



Your code should examine the name, password, and credit card number submitted, and verify that they are valid. A valid name is any non-empty string. A valid password is any string that is at least 6 characters long. A valid credit card number contains exactly 16 digits. Optionally, the credit card number can contain dashes between some or all groups of four digits. No other characters may be part of a credit card number. For example, the following are some examples of valid and invalid credit card numbers:

Valid	Invalid
1234567812345678 2457-1543-4367-4093 39485098-81902375 9834-34256678-9827	2457.1543.4367.4093 foo1234567812345678 12345678123456789 1234-5678-1234-5678- 12-34-56-78-12-34-56-78

Your PHP code's output should be a level-1 heading stating whether the data was valid or invalid, and a paragraph containing the data itself separated by commas. Replace the password by a string of * characters of equivalent length. Strip any dashes out of the credit card number while displaying it. For example, here are some outputs of your script for various form input:

Form Input	Output
name: Marty pw: booyah! cc: 1234-5678-1234-5678	**Successful.** Marty, *******, 1234567812345678
name: Kenneth pw: hulk cc: 11112222-33334444	**Denied! Invalid data.** Kenneth, ****, 1111222233334444
name: Jeff pw: quailman cc: 4321-4321x-4321-43210	**Denied! Invalid data.** Jeff, ********, 43214321x432143210

Use regular expressions for pattern matching and replacement.

Write your answer on the next page.

3. PHP (writing space)

4. JavaScript / DOM

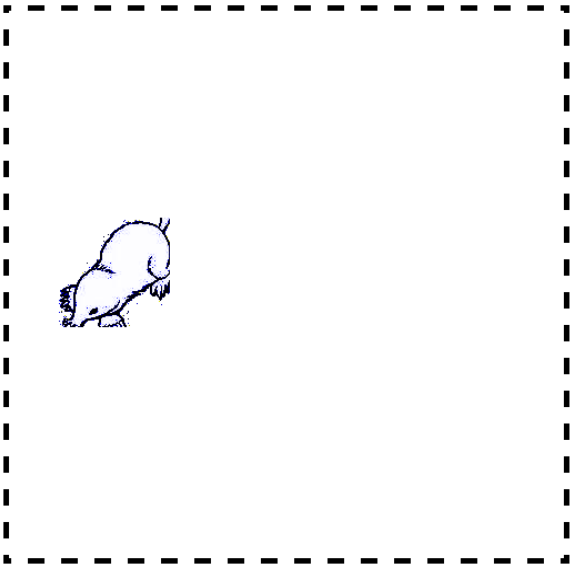
Write the Javascript code to add an image of a "ROUS" (rodent of unusual size) to the HTML `rodentarea` section below. (Assume that your code will be placed into a .js file that will be included by the HTML page.) The `rodentarea` is 500x500 px in size. The image comes from the file `rodent.gif` and is 100x100 px in size. The image is transparent other than the pixels of the rodent, so background colors behind it will show through.

Initially the image of the rodent appears at a random place within the `rodentarea` such that the rodent is entirely visible inside the area. When the user clicks on the rodent, the following occurs:

1. The background behind the image turns red for one second.
2. After the second has elapsed, the rodent jumps to a new random x/y position in the `rodentarea`, such that it is entirely visible inside the area.
3. After the rodent jumps to its new position, the background turns back to white.

```
#rodentarea {
    border: 5px dashed black;
    position: relative;
    width: 500px;
    height: 500px;
}
...
<div id="rodentarea"></div>
```

After your Javascript code runs, the appearance and behavior should be the following:



Write your answer on the next page.

4. **JavaScript / DOM (writing space)**

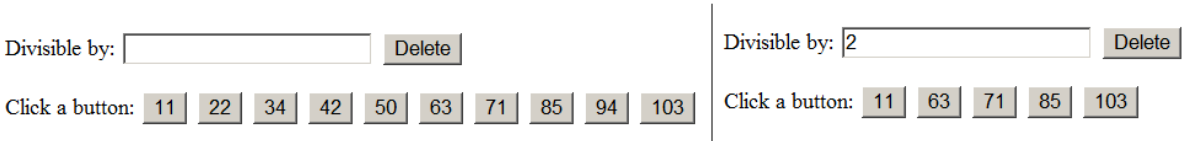
5. JavaScript / DOM

Write the Javascript code to accompany the following HTML code, so that when the Delete button is clicked, any button whose text value is divisible by the number written in the text field is removed from the page. You may assume that a valid number has been typed in the text field. The HTML code is the following:

```
<div id="controls">
    Divisible by:
    <input type="text" id="divisor" />
    <button id="del">Delete</button>
</div>

<div id="buttons">
    Click a button:
    <button>11</button> <button>22</button>
    <button>34</button> <button>42</button>
    <button>50</button> <button>63</button>
    <button>71</button> <button>85</button>
    <button>94</button> <button>103</button>
</div>
```

These screenshots show the initial page and its new appearance after typing 2 into the text field and pressing Delete:



Write your answer on the next page.

5. **JavaScript / DOM (writing space)**

6. Ajax / XML

Write the Ajax Javascript code to fetch and display XML data from the file named `movie.xml` (in the same directory as your code). This file contains lines spoken by a character in a movie. Your code should process the XML and display the character's lines, each in its own paragraph, in the format shown below. Assume that the code will execute on an HTML page containing a `div` with the `id` of `moviescene`, and insert the paragraphs into this `div`.

The XML data will be in a format that matches the following abbreviated example:

```
<character name="Captain Jack Sparrow" actor="Johnny Depp" />
     <line time="02:55">Captain Jack Sparrow, if you please, sir.</line>
     <line time="3:21">I'm in the market as it were.</line>
     ...
     <line time="2:41:38">On deck, you scabrous dogs! Man the braces!
     Let down and haul to run free. Now... bring me that horizon.  And
     really bad eggs?  Drink up, me 'earties, yo ho.</line>
</character>
```

For the XML data above, your code would produce the following content on the HTML page:

(02:55) Captain Jack Sparrow, if you please, sir.

(3:21) I'm in the market as it were.

(2:41:38) On deck, you scabrous dogs! Man the braces! Let down and haul to run free. Now... bring me that horizon. And really bad eggs? Drink up, me 'earties, yo ho.

7. SQL

Write an SQL query on the `imdb` database that will return the names of all characters that appeared in two or more of the Pirates of the Caribbean movies; that is, movies whose name begin with the substring "Pirates of the Caribbean". Ensure that the results are returned in alphabetical order. If it helps you, you may assume that the character is played by the same actor in both movies.

Recall the `imdb` database tables:

actors

id	first_name	last_name	gender
433259	William	Shatner	M
797926	Britney	Spears	F
831289	Sigourney	Weaver	F
...			

movies

id	name	year	rank
112290	Fight Club	1999	8.5
209658	Meet the Parents	2000	7
210511	Memento	2000	8.7
...			

roles

actor_id	movie_id	role
433259	313398	Capt. James T. Kirk
433259	407323	Sgt. T.J. Hooker
797926	342189	Herself
...		

directors

id	first_name	last_name
24758	David	Fincher
66965	Jay	Roach
72723	William	Shatner
...		

movies_directors

director_id	movie_id
24758	112290
66965	209658
72723	313398
...	

movies_genres

movie_id	genre
209658	Comedy
313398	Action
313398	Sci-Fi
...	

The following is a subset of the results returned:

```
+--------------+
| role         |
+--------------+
| Himself      |
| Marine       |
| Will Turner  |
+--------------+
```

If you join too many tables together that are not needed for the query, you will not receive full credit.
You should solve this problem using only the SQL syntax shown in class and the textbook.

X. Extra Credit

Write a haiku related to CSE 190M. A haiku is a Japanese poem made of three lines with 5, 7, and 5 syllables respectively. For example, the following is a haiku about Java:

public class Marty
public static void Booyah
System out println

(Any reasonable attempt at a haiku will get the +1 point.)