CSE 190 M, Spring 2008
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 this . Draw the image `geneva.jpg` as a smiley face occupying roughly 20% of the page width. It is possible that some CSS rules shown will not apply to any elements.

HTML:

```
<h1>Baby Geneva's Web Page</h1>

<div id="notice">
    <img src="geneva.jpg" alt="Geneva" />
    What is your name?
    <input type="text" size="20" value="Geneva" />
    <span class="notice">(Mine is cuter!)</span>
</div>

<p class="notice">I <br /> am <br /> 15 <br /> months <br /> old</p>
<p class="notice">
    My <br /> favorite <br /> toy <br /> is <br /> your <br /> cell phone
</p>

<h2>(written by Geneva, May 2008)</h2>
```

CSS:

```
h1, .notice {
    text-align: center;
    border: 2px solid black;
}
h2 { clear: both; }

h2, #notice {
    border-top: 1px dashed black;
    border-bottom: 1px dashed black;
}
```

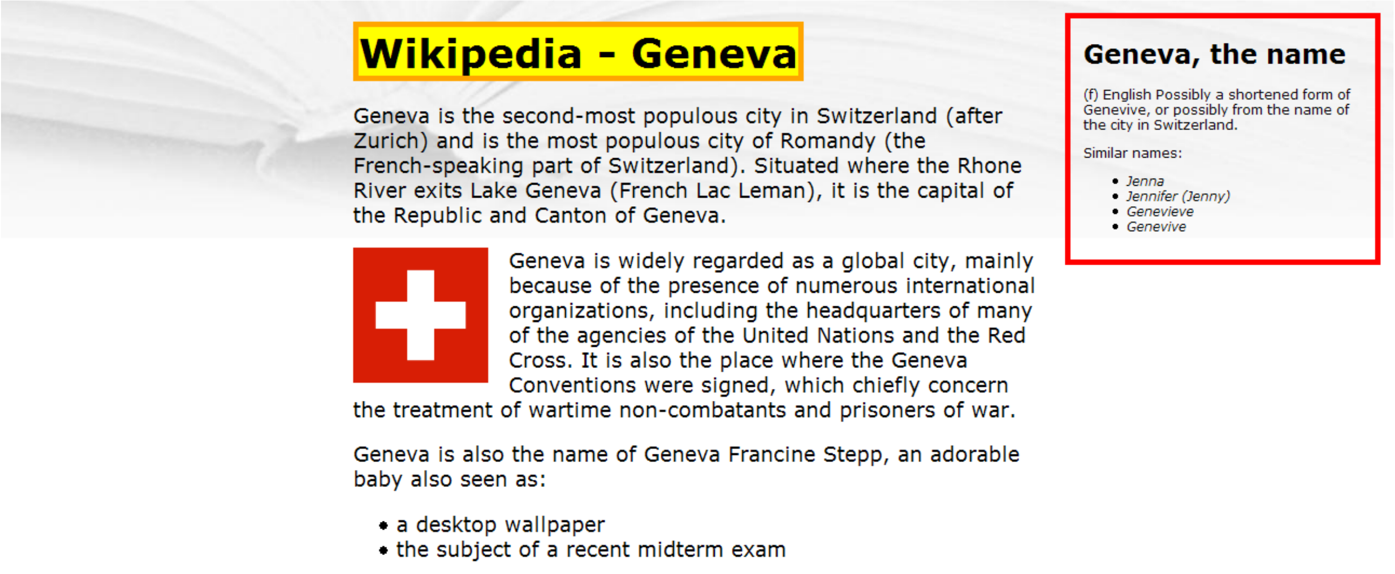
```
p.notice {
    border: 2px solid black;
    float: right;
    width: 10%;
}

#notice .notice {
    text-decoration: underline;
}
```

1. HTML / CSS Tracing (writing space)

2. HTML / CSS Coding

Write the HTML and CSS code necessary to recreate the following appearance onscreen. No manual line breaks have been inserted into the text.



Wikipedia - Geneva

Geneva is the second-most populous city in Switzerland (after Zurich) and is the most populous city of Romandy (the French-speaking part of Switzerland). Situated where the Rhone River exits Lake Geneva (French Lac Leman), it is the capital of the Republic and Canton of Geneva.

Geneva is widely regarded as a global city, mainly because of the presence of numerous international organizations, including the headquarters of many of the agencies of the United Nations and the Red Cross. It is also the place where the Geneva Conventions were signed, which chiefly concern the treatment of wartime non-combatants and prisoners of war.

Geneva is also the name of Geneva Francine Stepp, an adorable baby also seen as:

- a desktop wallpaper
- the subject of a recent midterm exam

Geneva, the name

(f) English Possibly a shortened form of Genevive, or possibly from the name of the city in Switzerland.

Similar names:

- *Jenna*
- *Jennifer (Jenny)*
- *Genevieve*
- *Genevive*

NOTE: 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 code that would pass the W3C validators. Assume that the given HTML text would appear inside the body of the page. Here are the details about the desired appearance:

- The overall page's text uses a 14pt Verdana font. The page has an overall background of `bg.jpg`, located in the same folder as the page; the background should appear only once and should not repeat.
- The text in the page's main `h1` heading has a yellow background and a solid 5px-thick orange border.
- The main section of page content is half as wide as the browser window and centered horizontally in the window. The text uses the default left-alignment.
- The image of the Swiss flag, `switzerland.png`, has 1em of horizontal space between itself and nearby text. Text wraps around the image as necessary.
- The area on the right about "Geneva, the name" appears 1em from the page's top/right corner. Its width should be 1/5 of the width of the overall browser window. It has a solid 5px-thick red border. Text in that part of the page uses a 9pt font. 1em separates the section's border from the content inside the section. The items in the bulleted list in this section are in italic.

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 the CSS styles on the page after next.

2. HTML / CSS Coding (writing space)

```
<h1>
    Wikipedia - Geneva
</h1>

<p>
    Geneva is the second-most populous city in Switzerland (after
    Zurich) and is the most populous city of Romandy (the French-
    speaking part of Switzerland). Situated where the Rhone River
    exits Lake Geneva (French Lac Leman), it is the capital of
    the Republic and Canton of Geneva.
</p>

<img src="switzerland.png" alt="Switzerland" />

<p>
    Geneva is widely regarded as a global city, mainly because of
    the presence of numerous international organizations, including
    the headquarters of many of the agencies of the United Nations
    and the Red Cross. It is also the place where the Geneva
    Conventions were signed, which chiefly concern the treatment
    of wartime non-combatants and prisoners of war.
</p>

<p>
    Geneva is also the name of Geneva Francine Stepp, an adorable
    baby also seen as:
</p>


<ul>
     <li>a desktop wallpaper</li>
     <li>the subject of a recent midterm exam</li>
</ul>


<h1>Geneva, the name</h1>

<p>
    (f) English

    Possibly a shortened form of Genevive, or possibly from
    the name of the city in Switzerland.
</p>


<p>
    Similar names:
</p>


<ul>
    <li>Jenna</li>
    <li>Jennifer (Jenny)</li>
    <li>Genevieve</li>
    <li>Genevive</li>
</ul>
```

2. HTML / CSS Coding (writing space)

3. PHP

Write a complete PHP web service baby.php that processes baby name data similar to that seen in the Baby Names assignment. The service reports the best (lowest) popularity ranking that name has ever held in the Social Security data. Your service accepts a GET request parameter named name. Its output is a single line of plain text containing the best ranking number, or -1 if no name parameter is passed or if the name is not found in the file. The input file to read, names.txt, is in the following format, with each line containing a baby's first name followed by some number of popularity rankings. The sample data below has 11 rankings per line, but for full credit your code should work regardless of how many rankings (1 or more) are on each line.

```
Martha 31 25 24 26 30 46 93 163 209 289 382
Martin 66 79 84 94 93 78 70 108 127 142 177
Martina 631 752 712 664 720 933 636 752 680 856 918
...
```

For example, if your service were requested as baby.php?name=Martha, its output would be 24, since that is Martha's best popularity ranking. If the request is baby.php?name=Martholomew, its output would be -1, since that name is not in the file. Your code should match case-insensitively; for example, the request baby.php?name=maRTIn should match the name Martin in the data file.

Write your answer on the next page.

3. PHP (writing space)

4. JavaScript / DOM

Write the JavaScript code to add event behavior to the following web page. Assume that your code will be placed into a .js file that will be included in the page's header.

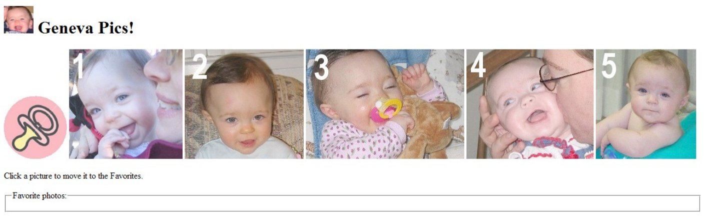
The page contains an area with five images of baby Geneva, and a "Favorites" area. Your code should arrange it so that when one of these five images is clicked, that image will be moved to the rightmost end of the `favorites` area. Also, when an image is moved, a bullet is added to the end of the `actions` list indicating this. The bullet's text should be, "Moved [image] to favorites." where [image] is equal to the `src` attribute of the image moved. (As you'll see in the output on the next page, even though the HTML's `src` attribute for each image is only a file name such as `geneva1.jpg`, when you access this information in your code, it will be a full URL. This is expected.) Nothing should happen when the user clicks on any other content on the page, such as text or other images such as the pacifier. Assume that the user won't click on an image that is already in the Favorites.

Move images by manipulating the order of elements in the DOM tree, not by manually setting x/y positions of elements. You may use Prototype's features if you like. You may *not* use `innerHTML` on this problem, nor change any HTML on the page; the only code you may add is JavaScript.

The following is the complete HTML code for the `body` of the page:

```
<h1>
    <img class="geneva" src="geneva_head.jpg" alt="Geneva" />  Geneva Pics!
</h1>
<div id="pics">
    <img src="pacifier.jpg" alt="pacifier" />
    <img class="geneva" src="geneva1.jpg" alt="Geneva" />
    <img class="geneva" src="geneva2.jpg" alt="Geneva" />
    <img class="geneva" src="geneva3.jpg" alt="Geneva" />
    <img class="geneva" src="geneva4.jpg" alt="Geneva" />
    <img class="geneva" src="geneva5.jpg" alt="Geneva" />
</div>
<p>Click a picture to move it to the Favorites.</p>
<fieldset id="favorites" class="geneva">
    <legend>Favorite photos:</legend>
</fieldset>
<ul id="actions"></ul>
```

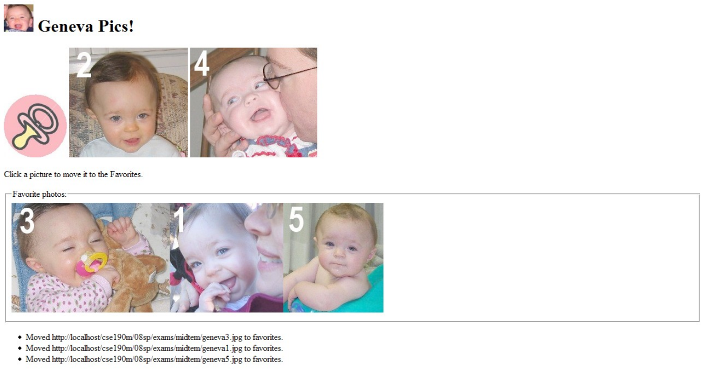
The following is the page's initial appearance and appearance after clicking images #1 and #5:



Geneva Pics!

Click a picture to move it to the Favorites.

Favorite photos:



Geneva Pics!

Click a picture to move it to the Favorites.

Favorite photos:

- Moved http://localhost/cse190m/08sp/exams/midtem/geneva3.jpg to favorites.
- Moved http://localhost/cse190m/08sp/exams/midtem/geneva1.jpg to favorites.
- Moved http://localhost/cse190m/08sp/exams/midtem/geneva5.jpg to favorites.

Write your answer on the next page.

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

5. JavaScript / DOM

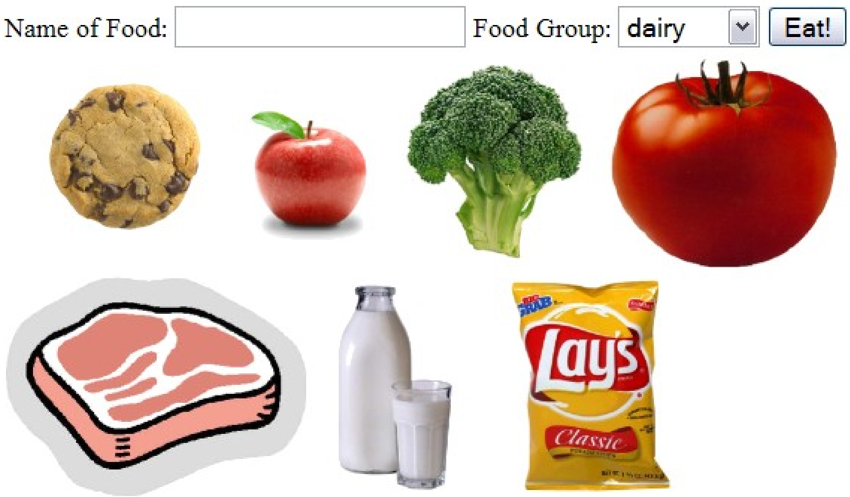
Write the Javascript code to add behavior to the following HTML code. The page contains a text input box with `id` of `foodname`, and a drop-down list with an `id` of `foodgroup`. The user types a name of a food item into the `foodname` text box, such as `apple` or `Cookie`, selects a food group from the drop-down list, such as `dairy` or `fruit`, and then clicks the button with `id` of `eat`. When the `eat` button is clicked, any element on the page that matches all of the following criteria will be removed from the page:

- The element is an `img` element that has a `class` of `food`.
- The element's food group matches the group chosen in the `foodgroup` drop-down list. Food groups are represented as `class` attributes. This is in addition to the `food` class; recall that a `class` attribute can specify multiple classes separated by spaces. For example, a jug of milk would have the following element:

  `<img src="milkjug.jpg" class="food dairy" alt="milk" />`
- The food item's name is the same as the text in the `foodname` box. The food's name is stored as the image's `alt` attribute. For example, if the user types `cookie`, `img` elements with an `alt` of `cookie` will be removed. Your code should be case-insensitive; for example, `coOKIe` should match images with an `alt` of `cookie`.

The relevant HTML code for the page is the following. Assume that Prototype is also included in the page. Your code should be general; it should not rely on the specific food images shown in the screenshot. You should also not rely on the exact ordering of the words in the `class` attribute.

```
<div>
    Name of Food: <input id="foodname" type="text" />
    Food Group:
    <select id="foodgroup">
        <option>dairy</option>
        <option>fats</option>
        <option>fruit</option>
        <option>meat</option>
        <option>veggies</option>
    </select>
    <button id="eat">Eat!</button>
</div>
<p>
    <img src="cookie.jpg" class="food fats" alt="cookie" />
    <img src="apple.jpg" class="fruit yummy food" alt="apple" />
    <img src="broc.jpg" class="food veggies" alt="broccoli" />
    <img src="tomato.jpg" class="fruit food" alt="tomato" />
    <img src="beefsteak.jpg" class="food meat" alt="steak" />
    <img src="milkjug.jpg" class="food dairy" alt="milk" />
    <img src="potatochips.jpg" class="fats food other" alt="chips" />
</p>
```



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 web service named `personality.php`, located in the same directory as your code. This code should run when the web page loads. When your code makes a GET request to this service without passing any parameters, its output is a set of XML data about users and their Keirsey personality types. Each user has a name, a 4-letter personality type, and a set of 70 answers to a personality survey. An answer can be an `A`, `B`, or `-` (dash) to indicate a blank response. (This data is similar to the data used in CSE 142's personality test homework assignment.) Answers may appear in either upper or lowercase. The XML data format matches the following abbreviated example:

```
<personalities>
  <person name="The frumious bandersnatch" type="INFP">
    <answers>-BBaBAA-BBbBBABBBBA-BaBBBBBbbBBABBBBBBBABB-BBBaBBABBBBBBB-BABBBBBBBBBBB</answers>
  </person>

  <person name="Minnie Mouse" type="ISTJ">
    <answers>BABA-AABABBBAABAABA-ABABAAAB-ABAAAAAA-AAAABAAABAAABAAAAAB-ABBAAAAAAAAA</answers>
  </person>
  ...
</personalities>
```

Your JavaScript code should fetch the list of persons and their types, and should turn each person's information into an item in an unordered list on the page with `id` of `output` that is already present on the page. Each list item should show the person's name, Keirsey type, and the count of how many total "B" (or "b") answers the person gave. For the XML data above, your code would produce the following HTML content (abbreviated with ...):

- The frumious bandersnatch: INFP (54 Bs)
- Minnie Mouse: ISTJ (18 Bs)
- ...

Write your answer on the next page.

6. Ajax / XML (writing space)

7. SQL

The Springfield Elementary school board is trying to crack down on grade inflation. To do this, they are trying to figure out which teachers are giving a lot of high grades in their courses. They have asked you to write an SQL query that can be run on the `simpsons` database that will find the names and course names of all teachers who gave 2 or more grades of C- or better in a given course. The query should show the teacher's name and the course name.

Recall the `simpsons` database tables:

students

id	name	email
123	Bart	bart@fox.com
456	Milhouse	milhouse@fox.com
888	Lisa	lisa@fox.com
404	Ralph	ralph@fox.com

courses

id	name	teacher_id
10001	Computer Science 142	1234
10002	Computer Science 143	5678
10003	Computer Science 190M	9012
10004	Informatics 100	1234

grades

student_id	course_id	grade
123	10001	B-
123	10002	C
456	10001	B+
888	10002	A+
888	10003	A+
404	10004	D+

teachers

id	name
1234	Krabappel
5678	Hoover
9012	Stepp

The results of the query would be the following, because Krabappel taught 142 and gave Bart a B- and Milhouse a B+, and Hoover taught 143 and gave Bart a C and Lisa an A+:

```
+-----------+----------------------+
| Krabappel | Computer Science 142 |
| Hoover    | Computer Science 143 |
+-----------+----------------------+
```

If a teacher taught more than one course with >= 2 high grades, your query should only show that teacher's name once along with any one of the courses in which the high grades were given by that teacher. Recall that a grade above C- is a string that occurs earlier than the string 'C-' in alphabetical order.

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

Choose a particular piece of web-related software, such as a particular web browser, operating system, or text editor. Then draw a picture of what a typical user of that software looks like. For example, you could draw a picture of a typical IE web surfer, a Mac user, a TextPad programmer, etc. Make sure to caption your drawing if necessary so that we know what software the person is using.

(Any drawing that appears to have taken more than a moment's work will get the +1 point.)