

CSE 154 Practice Final Exam
<Insert Your Favorite Day Here>

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, but closed notes. You may *not* use printed/written notes or 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 (...).
You must define any shorthand functions for selector functions (e.g. `document.getElementById`)
- You may not use JavaScript frameworks such as jQuery or Prototype when solving problems.
- 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!

('O')

Problem	Description	Earned	Max
1	HTML/CSS		*****
2	JS (DOM)		*****
3	JS / Ajax / JSON		*****
4	PHP		*****
5	Regex		*****
	SQL		*****
7	Short Answers		*****
X	Extra Credit		*****
TOTAL	Total Points		

1a. HTML

The following HTML code doesn't validate. Find and circle the errors with numbered labels, then list the required fixes for each error number below.

```
<!DOCTYPE HTML>
<html>
  /* Tips for exam */
  <head>
    <title>Tips for CSE 154 Final</title>
    <h2>How to Win</h2>
  </head>
  <body>
    <h1 id="tips">CSE 154 Exams - How to Win</h1>
    <ul>
      <p>Here are some tips:</p>
      <li class="tip">Sleep 8 hours night before</li>
      <li class="tip">Set an alarm</li>
      <li class="tip">Draw Pokemon for your TA</li>
    </ul>
    <div>
      
      <p id="important">Motivation puppy!</p>
    </div>
  </body>
</html>
```

1.b. CSS

Given the following HTML, write the CSS selectors and styling to fit the requirements.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Recipes R Us</title>
    <link type="text/css" rel="stylesheet" src="problem-2.css" />
  </head>
  <body>
    <h1>A Collection of the best recipes ever</h1>
    <div id="recipe-list">
      <ul>
        <li>Christmas Cookies</li>
        <li>Lasagna</li>
        <li>Ants on a Log</li>
        <li>Kimchi Burrito</li>
      </ul>
    </div>
    <div id="recipe-area">
      <h2>Christmas Cookies</h2>
      <div id="ingredients">
        <ul>
          <li>flour</li>
          <li>sugar</li>
          <li>magic</li>
        </ul>
      </div>
      <p class="instruction">Combine all the ingredients.</p>
      <p class="instruction">Wish on a shooting star.</p>
      <p class="instruction">Profit.</p>
    </div>
  </body>
</html>
```

Styling Requirements:

- The background of the entire page should be #123456.
- The color of the text for all the headings should be teal.
- Every element with the class instruction should have a border that is 2 pixels, dashed, and red.
- The div with the id recipe-list width should take up 40% of it's parent's width.
- The div with the id recipe-area width should take up 60% of it's parent's width.
- The text in the list inside the div with the id ingredients should have a font preference of Arial, Helvetica, or any other sans-serif font.

1.b. Write your solution below.

2. JavaScript / DOM

Write the **JavaScript code** to add behavior to the following page for keeping track of a to-do-list. The page UI allows the user to type an item into a text box. The user can click the "add" button to add the item to the bottom of the list. Each word in the phrase should be inserted as a `li`, inside a `ul` with the `id` of `list`.

If the user wishes to remove an item he or she can type the text of the item he or she wishes to remove in the text box and click the "remove" button. This should be case insensitive. For example, if the list only contains "foo" and the user tries to remove "FoO", it should be removed. If the user tries to remove an item that is in the list multiple times only the first occurrence should be removed.

The items should have background colors that alternate between white and yellow (first white, then yellow, then white, yellow, etc.). This should still be the case no matter how many items are removed or added and no matter what order these operations are done in. You **may not** use the CSS3 `nth-child` pseudo selector to do this.

The code should work for multiple clicks of the buttons. On each click it should clear any previous information you typed in the input boxes. **Do not use any JavaScript libraries such as jQuery or Prototype.** Here is the relevant HTML code for the page:

```
<h1>My super nifty to-do list</h1>
<ul id="list"></ul>
<div>
  <input type="text" id="item" />
  <button id="add">add</button>
  <button id="remove">remove</button>
</div>
```

These screenshots show the state after items have been added, and the state after items have been removed.

1. *Before anything has been added*

My super nifty to-do list

3. *After remove of item "go to the beach"*

My super nifty to-do list

- sleep
- buy cookies
- eat cookies
- go camping

2. *After 5 items added and none removed*

My super nifty to-do list

- sleep
- go to the beach
- buy cookies
- eat cookies
- go camping

4. *After remove of item "buy cookies"*

My super nifty to-do list

- sleep
- eat cookies
- go camping

Write your answer on the next page.

2. JavaScript / DOM (writing space)

3. Ajax/JSON

Suppose that there is a web service named `flights.php`, located on your web server in the same directory as your code. This service outputs JSON data describing flights between various cities. In this problem you will write Ajax JavaScript code to contact the web service (using a GET request), examine its JSON data, and display a list of possible flight prices and carriers.

The page contains two textboxes where the user can specify the start location and end location, a check box that they can check if they want to only see non-stop flights (flights with 0 stops) and a “Go!” button. When the button is pressed you should send an Ajax request passing the parameter of “start”. You can assume that the user has typed a valid location into each box before pressing the button

The JSON data returned by the web service consists of a list of end locations, each of which has a list of flights associated with it. The list of flights contains lists which each contain a price, a carrier and the number of stops.

```
{
  "Edinburgh":{
    "start":"Seattle",
    "flights":[{"carrier":"Delta","price":812,"stops":2},
              {"carrier":"Air France","price":1020,"stops":0},
              {"carrier":"Air France","price":1190,"stops":3}]
  },
  "New York":{
    "start":"Seattle",
    "flights":[{"carrier":"British Airlines","price":782,"stops":1},
              {"carrier":"Delta","price":1562,"stops":2},
              {"carrier":"United","price":957,"stops":1},
              {"carrier":"KLM","price":687,"stops":3},
              {"carrier":"KLM","price":1456,"stops":1}]
  }
}
```

The relevant existing HTML in the page is the following:

```
<div>
  <label>Start location: <input type="text" id="start" /></label>
  <label>End location: <input type="text" id="dest" /></label>
  <label>Non-stop? <input type="checkbox" id="stops"></label>
  <button id="go">Go!</button>
</div>
<div id="results"></div>
```

When the “Go!” button is clicked, clear previous results and read the JSON data with Ajax. Add a `h1` to the `results` div containing the text “Flights from” and then the start and destination locations. Turn each flight's data into a paragraph in the `results` div. In each paragraph, write the price of the ticket (with a “\$”) followed by the word “from”, the carrier's name and then the word “with”, the number of stops and the word “stops”. If the flight has a price below 1000 display its row in bold. For the example JSON shown above, the page is shown twice below, once with only non-stop flights and once with all flights.

Start location: End location: Non-stop?

Flights from seattle to Edinburgh

\$812 from Delta with 2 stops

\$1020 from Air France with 0 stops

\$1190 from Air France with 3 stops

Start location: End location: Non-stop?

Flights from seattle to Edinburgh

\$1020 from Air France with 0 stops

You may assume that the JSON data is valid in the format described previously, the data typed into the text boxes is valid, and that the `.php` service is reachable. **You may not use any Javascript libraries such as Prototype and JQuery.**

Write your answer on the next page.

3. Ajax/JSON (writing space)

4. Regular Expressions

a) Write a regular expression to match a **hexadecimal color code**. Remember, colors written in hexadecimal always start with a # and then contain 6 letters or numbers 0-9 and A-F. Letters can be lowercase or uppercase.

match:	don't match:
#000000	#1234567
#D3D3D3	4#D3D3D3
#abCDeF	#abCDeG

b) Write a regular expression to validate a **Mastercard number**. Mastercards have a 16 digit long number. The first number is always 5 and the second number is a 1, 2, 3, 4 or 5. The rest of the numbers can be anything. You should not look for or try to match dashes (-).

match:	don't match:
5112345678901234	51123456789012
5555555555555555	55555a55555555b55

c) Write a regular expression to match a **time** written like 11:04 AM. Times consist of an hour (1-12) followed by a colon, followed by minutes (00-59), followed by a space and then either AM or PM.

match:	don't match:
12:00 AM	12:0 AM
1:11 PM	14:00 PM
4:59 PM	0:20 AM
	02:00 AM
	4:60 PM

[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	\b	Any word boundary
(...)	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

5. PHP/JSON

Write the code for a web service `courses.php` that outputs JSON data about available courses at a school. This service should take two GET parameters named `start` and `end`, and search a data file for all courses that match those start/end times exactly and have open seats available.

Your PHP code will read a data file named `courses.txt`. Each line in that file matches the following format, with each token of information separated by a single space:

```
code startTime endTime seatsAvailable seatsTotal name
```

All tokens of data except the course name are guaranteed not to contain any spaces in them. You can assume all data in the file is valid, a number when you expect it to be a number and has no blank or malformed lines.

You may find an optional third parameter to the `split` function useful when writing your solution. If you pass `split` a number as a third parameter it will cap the number of times it splits to that number. Everything after the number of splits is passed will be placed in the last spot in the array. For example `split(":", "h:i:j:k", 3)` would return `{"h", "i", "j:k"}`.

The JSON that is output should contain a number labeled `count`. This should represent a count of all courses at the given start and end times. It should also contain a list called `courses` that contains a list for each course exactly matching the start and end times that has open seats. The list for each course should contain the code labeled as `code`, the number of seats left (the total seats minus the seats available) labeled as `seats`, and the name labeled as `name`,

You can assume both parameters will be passed. If no courses match the start/end and have seats you should send back the same data as usual, just leave the course array empty.

An example input file:

```
CSE154 130 230 250 250 Web Programming
CSE143 130 230 700 800 Computer Programming I
ANTH300 130 230 13 14 Anthropology
DANCE250 130 3 40 50 World Dance History
```

Output using the above input file and `courses.php?start=130&end=230`

```
{ "count":3,
  "courses":[ {"code" : "CSE143",
               "seats" : 100,
               "name" : "Computer Programming"
             },
             {"code" : "ANTH300",
               "seats" : 1,
               "name" : "Anthropology"
             }
            ]
}
```

5. PHP /JSON (additional writing space)

6. SQL

Write an SQL query to search the `world` database for all languages that are spoken as the official language of at least two "newly growing" countries. We will define a "newly growing" country as a country that has both of the following qualities: became independent after the year 1900, and contains at least one city with a population of over one million. For example, Malay would be listed because it is the official language of Malaysia which contains Kuala Lumpur (population 1,297,526) and became independent in 1957, and Indonesia which contains Jakarta (population 9,604,900) and became independent in 1945. Each language should be listed alphabetically and only once.

Recall the `world` tables:

code	name	continent	independence_year	population	gnp	head_of_state	...
AFG	Afghanistan	Asia	1919	22720000	5976.0	Mohammad Omar	...
NLD	Netherlands	Europe	1581	15864000	371362.0	Beatrix	...
...							

`countries`

country_code	language	official	percentage
AFG	Pashto	T	52.4
NLD	Dutch	T	95.6
...			

`languages`

```
+-----+
| name   |
+-----+
| Arabic |
| Chinese|
| English|
| French |
| German |
| Korean |
| Malay  |
| Russian|
| Spanish|
+-----+
9 rows in set (69 ms)
```

id	name	country_code	district	population
3793	New York	USA	New York	8008278
1	Los Angeles	USA	California	3694820
...				

`cities`

your
results at left.

When run on
`world` database,
query produces the

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.

7. Short Answer Questions

1. What keeps track of your cookies?
2. What line of code would be used in a php script to redirect to a page foo.php in the same relative directory?
3. Is `` an inline or block HTML element?
4. Why do we do validation of user input on the server side?
5. How do you get rid of a timer in JavaScript if you have a global variable timerID?
6. What does the tag `
` do?

X. Extra Credit

Draw a picture of your TA as a superhero.

(This is just for fun; any picture that appears to reflect more than a few moments' work will receive credit.)