CSE 154: Web Programming Spring 2018

Final Exam Solutions

Name: ______________________________

UWNet ID: ______________________________@uw.edu

TA (or section): ______________________________

Rules:

- You have 110 minutes to complete this exam.
- You will receive a deduction if you open the test before the start or keep working after the instructor calls for papers.
- This test is open-book and open note.
- You may not use any electronic or computing devices, including calculators, cell phones, smartwatches, and music players.
- Unless otherwise indicated, your code will be graded on proper behavior/output, not on style.
- Do not abbreviate code, such as writing ditto marks ("") or dot-dot-dot marks (...). You may not use JavaScript frameworks such as jQuery or Prototype when solving problems.
- You may use JavaScript function aliases like $ or qs **only if** you define them in your code.
- You may use the fetch syntax used in class in any JavaScript programs you write (you may assume checkStatus is included).
- 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.

Question	Score	Possible
0. HTML/CSS		10
1. JavaScript		20
2. JavaScript/Ajax		20
3. PHP Web Service		20
4. Regex		9
5. SQL		12
6. Short Answer		9
7. Extra Credit		
Total		100

0. CSS is to HTML as Swag is to Snowman

Solution:

```
<DOCTYPE! html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="index.css">
  </head>
  <body>
    <div id="top" class="circle">
      <div class="spot"></div>
      <div class="spot"></div>
    </div>
    <div id="middle" class="circle">
      <div class="spot"></div>
      <div class="spot"></div>
      <div class="spot"></div>
    </div>
    <div id="bottom" class="circle"></div>
  </body>
</html>
```

```
.circle {
  border: 2px solid black;
  border-radius: 50%;
  margin-left: auto;
  margin-right: auto;
}

#bottom {
  height: 200px;
  width: 200px;
}

#middle {
  height: 150px;
  width: 150px;
  display: flex;
  flex-direction: column;
  justify-content: space-around;
  align-items: center;
}

#top {
  height: 100px;
  width: 100px;
  display: flex;
  justify-content: space-around;
  align-items: center;

}
```

```
.spot {
  height: 20px;
  width: 20px;
  border-radius: 50%;
  background-color: black;
}
```

1. SNOW Day in Seattle!!!

Write a JavaScript program `snow.js` to add to the existing HTML and CSS (next page). This program animates a snow scene with falling snow.

Solution:

```
(function() {
  let boxH;

  window.onload = function() {
    setInterval(newSnowBall, 100);
    setInterval(fall, 50);
    // gets the height of #snow-scene
    boxH = parseInt(window.getComputedStyle(document.getElementById("snow-scene")).height);
  }

  function newSnowBall() {
    let div = document.createElement("div");
    let diameter = randomDiam();
    div.classList.add("snowball");
    div.style.left = randomNum(800 - diameter - 1) + "px"; // -1 is for right border
    div.style.top = 0 + "px";
    div.style.height = diameter + "px";
    div.style.width = diameter + "px";
    document.getElementById("snow-scene").appendChild(div);
  }

  function fall(div, speed, diameter) {
    let snowballs = document.querySelectorAll(".snowball");
    for(let i = 0; i < snowballs.length; i++){
      let snowball = snowballs[i];

      let diameter = parseInt(snowball.style.height);
      let speed = getSpeed(diameter);
      let yLoc = parseInt(snowball.style.top);

      yLoc += speed;
      if (yLoc < boxH) {
        snowball.style.top = yLoc + "px";
      } else {
        snowball.remove();
      }
    }

  }

  function getSpeed(diameter) {
    if (diameter == 30) {
      return 10;
    } else if (diameter == 20) {
      return 5;
    } else {
      return 2;
    }
  }

  function randomDiam() {
    let diams = [10, 20, 30];
    return diams[Math.floor(Math.random() * diams.length)];
  }

  function randomNum(maxValue) {
    return Math.random() * maxValue;
  }
})();
```

2. Fetching Pets Who Fetch with Fetch

Write a JavaScript file `pets.js` that plays a guessing game with the client, displaying the name and image of a random pet owned by a CSE 154 staff member, and which keeps track of how many times the client correctly guesses the TA/instructor who owns the pet.

Solution:

```
(function() {
  let ans;

  function $(id) {return document.getElementById(id);}

  window.onload = function() {
    dropDown();
    pet();
    $("guess-btn").onclick = guess;
  }

  function dropDown() {
    let url = "pets.php?mode=tas";
    fetch(url, {credentials: "include"})
      .then(checkStatus)
      .then(updateDD);
  }

  function updateDD(result) {
    result = result.split("\n");
    for (let i = 0; i < result.length; i++) {
      let tag  = document.createElement("option");
      tag.innerHTML = result[i];
      tag.value = result[i];
      $("ta-list").appendChild(tag);
    }
  }

  function pet() {
    let url = "pets.php?mode=random";
    fetch(url,  {credentials: "include"})
      .then(checkStatus)
      .then(JSON.parse)
      .then(updatePet);
  }

  function updatePet(resultJSON) {
    $("pet-name").innerHTML = resultJSON.petname;
    $("pet-type").innerHTML = resultJSON.type;
    let img = $("pet-img");
    img.src = resultJSON.images[Math.floor(Math.random() * resultJSON.images.length)];
    if (img.classList.contains("hidden")) {
      img.classList.remove("hidden");
    }
    $("guess-btn").classList.remove("disabled");
    ans = resultJSON.name;
  }

  function guess() {
    $("guess-btn").classList.add("disabled");
    var dropDown = $("ta-list");
    var guess = dropDown.value;
    if (guess == ans) {
      let correct = $("correct").innerHTML;
      correct = parseInt(correct) + 1;
      $("correct").innerHTML = correct;
    }
    let total = $("total").innerHTML;
    total = parseInt(total) + 1;
    $("total").innerHTML = total;
    pet();
  }
})();
```

3. The Thing That We Fetch From For Fetching Pets Who Fetch

Write a PHP web service called pets.php which provides data about CSE 154 staff and their pets. For this problem, assume your PHP file is in the same directory as pets.txt and a collection of folders for each TA/instructor.

Solution:

```
<?php
  $pets = file("pets.txt");

  if(isset($_GET["mode"]) &&  $_GET["mode"] == "tas") {
    header("Content-type: text/plain");
    $final_array = array();
    foreach($pets as $line) {
      $ta_array = explode(" ", $line, 4);
      if(!in_array($ta_array[0], $final_array)) {
        array_push($final_array, $ta_array[0]);
        print "$ta_array[0]\n";
      }
    }
  } else if(isset($_GET["mode"]) && $_GET["mode"] == "random") {
    $pet = $pets[array_rand($pets)];
    list($name, $petname, $type, $age) = explode(" ", $pet, 4);
    $images = glob("$name/$petname/*.jpg");
    foreach($images as $image) {
      $image = str_replace("$name/$petname/", "", $image);
    }
    $output = [
      "name" => $name,
      "petname" => $petname,
      "type" => $type,
      "age" => $age,
      "images" => $images
    ];
    header("Content-type: application/json");
    print json_encode($output);
  } else {
    header("HTTP/1.1 400 Invalid Request");
    header("Content-type: text/plain");
    print "Error: Please pass in a mode parameter of tas or random.";
  }
?>
```

4. RegEx

a. Caaafffeeeinee!!!!!

It's finals week, and as true Seattlites, we need caffeine to survive it. Write a regular expression to match all Strings that contain the words "coffee", "mocha", or "green tea" containing no digits or special characters (space characters should be accepted). Both lower-case and upper-case letters are allowed for any letter in the String.

Solution:

```
^[a-zA-Z\s]*(coffee|mocha|green tea)[a-zA-Z\s]*$
```

b. Deoxy-what?

Write a regular expression to match DNA sequences that consist of codons (sequences of three nucleotides [A, C, T, or G]). The expression should only match sequences that start with the start codon 'ATG' and end with one of the following three stop codons: 'TAA,' 'TAG,' or 'TGA.' Any number of codons with any sequence of nucleotides may be between a start codon and a stop codon. Sequences that do not have complete codons (i.e., those which do not have nucleotide counts that are multiples of three) should not be accepted.

Solution:

```
^ATG([ACGT]{3})*((TAA)|(TAG)|(TGA))$
```

c. SELECT-em ALL!

Write a regular expression that accepts simple SQL select statements. That is, any SQL statement that:

- starts with "SELECT" followed by one or more comma-separated column names followed by
- by a single "FROM" statement followed by
- one or more comma-separated table names and
- is finished optionally with an semi-colon (;)

A valid column or table name contains only letter characters (no spaces, numerical, or special characters), except that a column name may optionally include a single '.' within the name string (not at the start or end). SELECT and FROM must be **both** all-lowercase or **both** all-uppercase.

Solution:

```
^(select ([a-zA-Z]*\.?[a-zA-Z]*)(, [a-zA-Z]*\.?[a-zA-Z]*)* from [a-zA-Z]*;?)|
(SELECT ([a-zA-Z]*\.?[a-zA-Z]*)(, [a-zA-Z]*\.?[a-zA-Z]*)* FROM [a-zA-Z]*;?)$
```

5. Would You Like a Table With That Menu?

Consider the following tables in a database called "Cafe":

Menu

mid	item	price	cost	category	subcategory
1	Brewed Coffee (Black)	1.25	0.65	Drink	Coffee
2	Espresso	3.75	1.15	Drink	Coffee
3	Jeremy's Unicorn Frappe	8.00	2.75	Drink	Coffee
4	Bubble Tea	3.00	1.75	Drink	Tea
5	Blueberry Muffin	3.00	1.20	Bakery	Muffin
6	Honey Toast	4.00	1.85	Bakery	Bread
7	Chocolate Chip Cookie	1.25	0.50	Bakery	Cookie
8	Chocolate Chip Cookie (Gluten-Free)	1.25	0.60	Bakery	Cookie
...	...	...	...	...	...

Customers

cid	firstname	lastname
153	Sam	Kaufman
154	Kyle	Thayer
155	Jeremy	Zhang
156	Kelley	Chen
157	Melissa	Medsker
...	...	...

Orders

oid	mid	cid	date	time	qty
1000	2	157	2017-11-29	03:00	154
1001	3	155	2017-12-02	13:00	1
1002	3	155	2017-12-08	13:30	50
1003	4	156	2017-12-09	09:30	4
1004	1	153	2017-12-10	01:00	12
...	...	...	...	...	...

The Menu table has a primary key of **mid**, the Customers table has a primary key of **cid**, and the Orders table has a primary key of **oid**. The **mid** and **cid** attributes in Orders correspond to the primary keys in Menu and Customers, respectively.

The questions are on the next page.

(a) Write a SQL query which returns the item name, date, time, and the last name of the customer for all Coffee Drink orders (Drink orders with subcategory 'Coffee') which were made in the year 2017. Results should be ordered by the most recent order first.

Hint: You can treat the `date` attribute as a normal SQL string to determine whether it is a date in 2017. `date` and `time` are comparable as expected. That is, the date `2017-12-08` (December 8, 2017) is considered 'greater than', or 'more recent' than `2017-05-02` (May 2nd, 2017) and the time `13:00` (13:00 or 1:00 PM) is considered 'more recent' in a given day than `01:00` (01:00 or 1:00 AM).

Solution:

```
SELECT i.item, o,date, o.time, c.lastname
FROM Orders i
JOIN Customers c ON i.cid = c.id
JOIN Menu m ON i.mid = m.id
WHERE i.date >= '2017-01-01'
AND i.date <'2018-01.01'       // optional
AND m.category = 'Drink'       // optional
AND m.subcategory = 'Coffee'
ORDER By i.date DESC, i.time DESC;
```

(b) Write a SQL query which returns the first and last name of the customer, time of order, menu item, and category for all customers who have both an order for a Drink and an order for a Bakery item in the database (they may have more than one order for both categories, but you should include all of the orders for customers who meet this requirement). There should be no duplicate rows with the same customer, order time, menu item, and category.

Solution:

```
SELECT DISTINCT c.firstname, c.lastname, o1.time, m1.item, m1.category
FROM Orders o1, Customers c, Menu m1, Orders o2, Menu m2
WHERE o1.mid = m1.mid
AND o2.mid = m2.mid
AND c.cid = o1.cid
AND c.cid = o2.cid
AND ((m1.category = "Drink" AND m2.category = "Bakery") OR
    (m1.category = "Bakery" AND m2.category = "Drink"))
```

6. Short Answers

1. What is the difference between a relative link and an absolute link?

   Solution:

   A relative link is relative to the path of the file that references the link. An absolute link is the full url, including the http/https at the beginning.

2. Why is it important to use the module pattern in JavaScript?

   Solution:

   To prevent pollution of the global namespace; to prevent conflict between duplicate variables/functions between different scripts.

3. Provide one real-world example where a cookie would be more appropriate to use than a session

   Solution:

   Client-side navigation preferences for an e-commerce shopping site.

4. Provide one real-world example where a session would be more appropriate to use than a cookie

   Solution:

   Server-side login validation on a banking site.

5. What is one advantage of using a SQL database over text files to store data?

   Solution:

   SQL databases are easier to query, join, manipulate, and are easier to scale than text-files. It is also much easier to allow multiple users to query/update the same database at once.

6. What are two ways of making a website more accessible for users with different abilities?

   Solution:

   Using colors that are easily distinguishable for users who are color-blind. Avoiding the use of directional text. Using attributes like descriptive alt text which can be easily identified by screenreaders.

7. What is the difference between a GET and POST request in PHP?

   Solution:

   A GET request is used primarily to retrieve data from the server, and may include parameters in the request URL. A POST request is primarily used to send data to the server (data may still be returned in the response) and any parameters sent are encrypted (e.g. using FormData).

8. Why is it important to specify the content type header in PHP when outputting something?

Solution:

By default, PHP outputs content as HTML. Content type needs to be set explicitly whenever it is expected in a different format (e.g. plain text, JSON, etc.)

9. Explain one way someone could do something bad by exploiting a vulnerability in your site if it wasn't written securely.

Solution:

Injecting a JS script onto the page when the page is loaded, or when code to do so is entered in a form input. Submitting a form with SQL modification queries (e.g. DROP TABLE) when there is no client-side verification/escaping.

7. Extra Credit (1 extra credit point)

If your TA were an HTML tag, which one would they be, and why? You may also give us an artistic rendering of your TA as an HTML tag, if you prefer. (Drawing, poetry, etc). Any work that appears to have taken more than 1 minute of effort and is not offensive/inappropriate will receive the extra credit point.