University of Maryland College Park
Department of Computer Science
CMSC335 Spring 2023
Exam #1Key

FIRSTNAME, LASTNAME (PRINT IN UPPERCASE):

STUDENT ID (e.g., 123456789):

Instructions

- This exam is a closed-book, closed-notes exam with a duration of 75 minutes and 200 total points.
- You may lose credit if you do not follow the instructions below.
- **At this point, you must write your name and id at the top of this page and add your directory id (e.g., terps) at the end of odd-numbered pages.**
- Please use a pencil or erasable pen to answer the exam.
- **Do not remove the exam's staple or bend any pages, as doing so will interfere with the scanning process.**
- **Provide answers in the rectangular areas (boxes).** Make a note if you continue a problem on another page(s). **For multiple-choice questions, please fill in the bubble (do not circle).**
- For multiple-choice questions, you can assume only one answer unless stated otherwise.
- **Your code must be efficient and as short as possible.**
- You don't need to use meaningful variable names; however, we expect good indentation.
- **You must stop writing once the time is up; otherwise, you will lose significant credit.**

Grader Use Only

Problem #1 (Miscellaneous)	68	
Problem #2 (CSS)	20	
Problem #3 (HTML)	32	
Problem #4 (Form)	24	
Problem #5 (JavaScript Code)	56	
Total	200	

Problem #1 (Miscellaneous)

1. (3 pts) Based on our class discussion, which of the following allows us to have more than one server running at a particular domain?
   - (a.) Port numbers
   - (b.) NaN
   - (c.) localhost
   - (d.) None of the above.

   Answer: a.

2. (3 pts) In a web server, **127.0.0.1** corresponds to:
   - (a.) htdocs
   - (b.) localhost
   - (c.) superhost
   - (d.) None of the above.

   Answer: b.

3. (3 pts) In the Apache server provided by XAMPP, the documents we want the webserver to deliver are in the directory:
   - (a.) htdocs
   - (b.) localhost
   - (c.) index.html
   - (d.) None of the above.

   Answer: a.

4. (3 pts) A URL (Uniform Resource Locator):
   - (a.) It is a subcategory of URI that provides information on how to access a resource.
   - (b.) It is a subcategory of URI that does not provide information on how to access a resource.
   - (c.) It is exactly the same as a URI.
   - (d.) None of the above.

   Answer: a.

5. (3 pts) From what we discussed in the lecture, if we can see a page when we specify an URL that ends with the name of a directory (e.g., https://www.cs.umd.edu/class/spring2023/cmsc000 where cmsc000 is a directory), then:
   - (a.) The directory has a text file called README.txt.
   - (b.) The directory has a file called htdocs.
   - (c.) The directory has a file called index.html or index.shtml.
   - (d.) None of the above.

   Answer: c.

6. (3 pts) An HTTP **post** request:
   - (a.) **Can** be bookmarked in the browser.
   - (b.) Includes parameters in the URL.
   - (c.) Usually modifies the state of the server.
   - (d.) None of the above.

   Answer: c.

7. (3 pts) An HTTP **get** request:
   - (a.) **Cannot** be bookmarked in the browser.
   - (b.) Includes parameters in the URL.
   - (c.) Usually modifies the state of the server.
   - (d.) None of the above.

   Answer: b.

8. (3 pts) An HTML attribute can appear:

   a. In the start tag.
   b. In the end tag.
   c. a. and b.
   d. None of the above.

   Answer: a.

9. (3 pts) The DOM (Document Object Model) represents:

   a. A document using a list.
   b. A document using a tree.
   c. A document using a set.
   d. A document using a stack.

   Answer: b.

10. (3 pts) The output of the following code fragment is:

```
let answer = (Window.NaN == Number("Rose"));
document.writeln(answer);
```

   a. true
   b. false
   c. undefined
   d. null

   Answer: b.

11. (3 pts) Which of the following should you use if you need to apply the same CSS style to several elements (e.g., paragraphs, tables)?

   a. A class selector.
   b. An id selector.
   c. A pseudo-class selector.
   d. None of the above.

   Answer: a.

12. (3 pts) JavaScript conforms to which of the following specifications?

   a. ECMAScript
   b. JScript
   c. W3C Validator
   d. None of the above.

   Answer: a.

13. (3 pts) The output of the following code fragment is:

```
document.writeln(parseInt("36.9feet"));
```

   a. 36.9
   b. 36
   c. NaN
   d. undefined

   Answer: b.

14. (3 pts) The functions Window.NaN() and Number.NaN() always generate the same results.

   a. true
   b. False

   Answer: b.

15. (3 pts) Name three primitive data types in JavaScript:

Answer: Any three from null, boolean, number, string, undefined, bigint or symbol

16. (4 pts) Complete the following assignment so **x** is assigned a random floating-point value between -51 (exclusive) and -1 (inclusive).

```
let x =
```

Answer: `-1 * ((Math.random() * 50) + 1);`

17. (4 pts) Rewrite the following assignment using template literals. **Make sure your answer is as short as possible.**

```
let answer = "Dear<em>" + user + "(\"Boss\")</em>";
```

Answer: `` `Dear<em>${user}("Boss")</em>` ``;

18. (15 pts) Complete the implementation of the **sortStudents** function below. The function will sort an array of objects. Each object has a **name** (string) and an **id** attribute (integer). If the **byName** parameter is true, the array will be sorted by name in alphabetical order (using the localCompare function); if **byName** is false, the array will be sorted by **id**. The following is an example of an array of objects.

```
let students = [{name: "John", id: 3}, {name: "Peter", id: 2}];
```

Your function must work for different kinds of arrays (not just the previous array). Note: In JavaScript, a function (inner) within another function (outer) can access the variables and parameters of the containing function (outer). You must use the JavaScript sort() function (and not define a sorting function).

```
function sortStudents(students, byName) {
```

Answer:

```
function sortStudents(students, byName) {
   students.sort(function(x,y) {
     if (byName) {
        return x.name.localeCompare(y.name);
     } else {
        return x.id - y.id;
     }
   });
}
```

Problem #2 (CSS)

1. (9 pts) Define a CSS rule that makes the div elements size **three rem**, the background color **blue,** and the text color **red.**

Answer: `div {font-size: 3rem; background-color: blue; color: red}`

2. (3 pts) Define a CSS rule, so the color of paragraphs becomes green when we hover over them.

Answer: `p:hover { color:green }`

3. (4 pts) Define a CSS rule for the following HTML based on an **id selector** that makes the text color of **pre** tags yellow.

```
<pre id="loc">
   Location
</pre>
```

Answer: `#loc { color: yellow }`

4. (4 pts) Define a CSS rule for the following HTML based on a **class selector** that makes the font size **two em**.

```
<span class="in">Intro</span>
```

Answer: `.in {font-size: 2em }`

Problem #3 (HTML)

1. (5 pts) Using the <img> tag, define an image entry where the image name is **door.png,** and the message "a door" will appear when the image cannot be displayed.

Answer:

```
<img src="door.png" alt="a door">
```

2. (13 pts) In the rectangular area below, write the HTML that goes inside of the <body></body> tags that will generate the following list. Notice that the first entry of the main list has a list. You may not type any numbers as part of your HTML.

1. Bio
    - Lab
    - OH
2. Dinner

Answer:

```
<ol>
    <li>Bio
        <ul>
            <li>Lab</li>
            <li>OH</li>
        </ul>
    </li>
    <li>Dinner</li>
</ol>
```

3. (14 pts) In the rectangular area below, write the HTML that goes inside of the <body></body> tags that will generate the following table. To create the border, use the table **border** attribute with a value of one. Using an inline CSS approach, make the size of the header two rem. **You may not add any CSS file.**

Sem
Fall

Answer:

```
<table border="1">
    <tr><th style="font-size:2rem">Sem</th></tr>
    <tr><td>Fall</td></tr>
</table>
```

Problem #4 (Form)

Define the form (just the form tags and its contents) that will call the script **score.php** using the **get** method when the user clicks on the "Get Score" button. The value one will appear in the text field by default, and it is the value that will appear in the text field when the user clicks on the "Reset" button. Make sure you add the text Project# to the form.

Project# 1 Get Score Reset

Answer:

```
<form action="score.php" method="get">
    Project# <input type="text" value="1">  <!-- We are ignoring the name attribute →
    <input type="submit" value="Get Score">
    <input type="reset">
</form>
```

Problem #5 (JavaScript Code)

Implement a JavaScript function called **getHistogram** that returns HTML with a horizontal histogram based on the parameters **values** and **symbol**. The **values** parameter is a string that includes the length of each histogram's row. For example, the string "4-2-6", represents a histogram where the first row has a length of 4, the second 2, and the last 6. The **symbol** parameter represents the symbol to use in the histogram. For example, calling **getHistogram("4-2-6", "*")** will return the following string:

```
<pre>
****
**
******
</pre>
```

For this problem:

1. You can assume a dash will separate the values in the **values** parameter.
2. If one of the values in the **values** string is not a number (e.g., "5-X-8" has an invalid second value), then the row associated with the invalid value will have the string "**invalid value**".
3. The string returned by the function must rely on the <pre></pre> tags.
4. Remember, you can use the split function to turn a string into an array.
5. **You may not write auxiliary functions.**

Answer:

```
function getHistogram(values, symbol) {
  let data = values.split("-");
  let answer ="<pre>";

  for (let size of data) {
    if (isNaN(Number(size))) {
       answer += "invalid value\n";
    } else {
      for (let i = 1; i <= size; i++) {
        answer += symbol;
      }
      answer += "\n";
    }
  }

  return answer += "</pre>";
}
```