



University of Maryland College Park

Dept of Computer Science

CMSC389N Spring 2017

Midterm II Key

Last Name (PRINT): _____

First Name (PRINT): _____

University Directory ID (e.g., umcpturtle) _____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 200 points.
- The exam is a 75 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- **You may not use jQuery nor Bootstrap.**
- **You don't need to use meaningful variable names; however, we expect good indentation.**

Grader Use Only

#1	Problem #1 (HTML/CSS/JS Language)	(90)	
#2	Problem #2 (JavaScript Coding/Custom Types)	(50)	
#3	Problem #3 (JavaScript Coding/Dynamic HTML)	(60)	
Total	Total	(200)	

Problem #1 (HTML/CSS/JS Language)

1. (3 pts) In JavaScript which value is associated with object properties that do not exist?
- a. undefined
 - b. null
 - c. 0
 - d. None of the above.

Answer: a.

2. (3 pts) Which of the following expressions are true in JavaScript?
- a. **NaN == NaN**
 - b. **NaN === NaN**
 - c. true
 - d. None of the above.

Answer: c

3. (3 pts) Which of the following expressions are true in JavaScript? Circle all the ones that are true.
- a. new Object(false)
 - b. new Object(true)
 - c. "20" === 20
 - d. None of the above.

Answer: a, b

4. (3 pts) What is the output of the following program (notice it relies on let)?

```
let x = 25;

if (x > 5) {
  let x = 100;
  document.writeln("First: " + x + "<br>");
}
document.writeln("Second: " + x + "<br>")
```

Answer:

First: 100
Second: 25

5. (3 pts) What is the output of the following program (notice it relies on var)?

```
var x = 25;

if (x > 5) {
  var x = 100;
  document.writeln("First: " + x + "<br>");
}
document.writeln("Second: " + x + "<br>")
```

Answer:

First: 100

Second: 100

6. (3 pts) Which of the following allow us to retrieve the character associated with an index position in a string? Circle all that apply.

- a. `charAt()`
- b. `[]`
- c. `{}`
- d. None of the above.

Answer: a, b

7. (3 pts) What is the output of the following code fragment?

```
let s1 = Symbol("Test2"), s2 = Symbol("Test2"), s3 = Symbol("Test3");
if (s1 === s2) { document.writeln("S1<br>"); }
if (s1 === s3) { document.writeln("CASE<br>"); }
document.writeln("DONE");
```

Answer: DONE

8. (3 pts) What is the difference between `Global isNaN()` and `Number.isNaN()`?

- a. There is no difference; they are equivalent.
- b. `Global isNaN` tries to convert the argument to a number.
- c. There are two unrelated functions.
- d. None of the above.

Answer: b

9. (3 pts) What is the output of the following statement?

```
document.writeln("F: " + typeof new Array(2) + "<br>");
```

- a. F: object
- b. F: Array
- c. F: undefined
- d. None of the above

Answer: F: object

10. (3 pts) Which of the following will always work when trying to identify an object as an array?

- a. `typeof`
- b. `instanceof`
- c. `Array.isArray()`
- d. `isNaN()`
- e. None of the above

Answer: c

11. (3 pts) Which of the following objects allow us to issue an AJAX request?

- a. `XMLHttpRequest`
- b. `Object`
- c. `JSONObject`
- d. None of the above

Answer: a

12. (4 pts) The function **stats** has the following prototype:

```
function stats(age, enterCollege, finishedCollege, school)
```

Write a function call that will use the spread operator and the array [1990, 1994] in order to initialize the `enterCollege` and `finishedCollege` parameters. You can assume the `age` and `school` parameters are 20, and "UMCP", respectively.

Answer: `stats(20, ...[1990, 1994], "UMCP");`

13. (4 pts) Rewrite the second assignment using template literals.

```
let name = "Richardson";  
let introduction = "Dear Prof. " + name;
```

Answer: `let introduction = `Dear Prof. ${name}`;`

14. (4 pts) What is the output of the following code fragment?

```
let data = new Array(4);  
data[0] = "red";  
data[3] = "blue";  
document.writeln(data[0] + ", " + data[3] + "<br>");  
data.length = 2;  
document.writeln(data[0] + ", " + data[1] + "<br>");
```

Answer:

red, blue
red, undefined

15. (4 pts) Using a single `document.writeln` print **true** if the first and last character of the string associated with a variable called **name** are the same, and **false** otherwise. You may not use conditionals (i.e., if statements).

One Possible Answer: `document.writeln(name[0] == name[name.length - 1]);`

16. (4 pts) Using the `=>` operator initialize the variable **myFunc** with a function that takes no parameters and returns the value 400.

```
let myFunc =
```

Answer: `() => 400`

17. (6 pts) Provide an expression that relies on `Math.random()` that assigns to `x` a random integer value in the range [1, 20]. Notice the range includes 1 and 20.

```
let x =
```

Answer: `let x = Math.floor(20 * Math.random()) + 1`

18. (8 pts) Complete the assignment below so the reversed string ("30, 20,10") is assigned to **result**. You may only use a single assignment otherwise you will not receive credit.

```
let line = "10, 20, 30";  
let result = YOUR CODE HERE
```

One Possible Answer: `line.split(",").reverse().join(",");`

19. (8 pts) Print the contents of the following map using a **for of** loop and `document.writeln`. The expected output is:

```
Course: cmsc132, Value: 4  
Course: cmsc389N, Value: 3
```

```
let map = new Map();  
map.set("cmsc132", 4);  
map.set("cmsc389N", 3);
```

One Possible Answer:

```
for (let [key, value] of map) {  
    document.writeln("Course: " + key + ", Value: " + value + "<br>");  
}
```

20. (8 pts) Using the `sort` array function and anonymous functions, define the function used by `sort` that will sort the elements of the **students** array by increasing gpa value.

```
let students = [  
    { name: "Kelly", gpa: 3.7},  
    { name: "Sam", gpa: 4.0},  
    { name: "Peter", gpa: 2.3}  
];
```

students.sort(YOUR FUNCTION HERE)

One Possible Answer: `students.sort(function (x, y) { return x.gpa - y.gpa; });`

21. (7 pts) Define your own Error type called **InvalidPasswordError**. The following is an example of the using your Error type.

```
try {
  let code = prompt("Enter password");
  if (code != "terps")
    throw new InvalidPasswordError("Invalid password");
  document.writeln("Welcome<br>");
} catch(error) {
  alert(error.message);
}
```

Answer:

```
function InvalidPasswordError(message) {
  this.message = message;
}
InvalidPasswordError.prototype = new Error();
```

Problem #2 (JavaScript Coding/Custom Types)

Write JavaScript (NOT PHP) that defines two “classes” using the approach presented in class. **If you use E6 class definitions (similar to what you have in Java) you will not receive any credit for this problem.**

1. Game

- Define a Game “class” that has two **private** variables named **name** and **price**.
- Define a constructor that has two parameters: name and price.
- Define two get methods, getName and getPrice, that return the name and price, respectively.
- The Game class has a prototype that defines a defaultLanguage variable with the value “English” and an overview() method that prints the name and price (see example below for format information).

2. VideoGame

- Define a VideoGame “class” that “extends” the Game class. The class has a **private** variable named memoryRequirements.
- Define a constructor that has three parameters: name, price, and memoryRequirements.
- Define a get method named getMemoryRequirements that returns the memoryRequirements.
- You need to define the appropriate prototype for this “class”.

The following is an example of using the “classes” you will define.

```
main();

function main() {
    let game1 = new Game("Chess", 14.95);

    /* Game */
    document.writeln("Overview<br>");
    game1.overview();
    document.writeln("End Overview<br><br>");
    document.writeln("Game Name: " + game1.getName() + "<br>");
    document.writeln("Game Size: " + game1.getPrice() + "<br>");
    document.writeln("Game Language: " + game1.defaultLanguage + "<br><br>");

    let videoGame1 = new VideoGame("Netrix", 15.95, 4000);
    document.writeln("VGame Name: " + videoGame1.getName() + "<br>");
    document.writeln("VGame Price: " + videoGame1.getPrice() + "<br>");
    document.writeln("VGame Language: " + videoGame1.defaultLanguage + "<br>");
    document.writeln("VGame Memory Requirements: " + videoGame1.getMemoryRequirements() + "<br>");
    videoGame1.overview();
}
```

Output

Overview

Name: Chess, Price: 14.95

End Overview

Game Name: Chess

Game Size: 14.95

Game Language: English

VGame Name: Netrix

VGame Price: 15.95

VGame Language: English

VGame Memory Requirements: 4000

Name: Netrix, Price: 15.95

Game Class Answer:

```
function Game(name, price) {
    this.getName = function() {
        return name;
    }
    this.getPrice = function() {
        return price;
    }
}

Game.prototype = {
    constructor: Game,
    defaultLanguage: "English",
    overview: function() {
        document.write("Name: " + this.getName());
        document.write(", Price: " + this.getPrice() + "<br>");
    }
};
```

VideoGame Class Answer:

```
function VideoGame(name, price, memoryRequirements) {
    Game.call(this, name, price);

    this.getMemoryRequirements = function() {
        return memoryRequirements;
    }
}

VideoGame.prototype = Object.create(Game.prototype);
VideoGame.prototype.constructor = VideoGame;
```


Problem #3 (JavaScript Coding/Dynamic HTML)

Write a **JavaScript (NOT PHP)** program that allow us to display in a table links to webpages. The links will be generated based on following array that will be a global variable:

```
let data = ["syllabus.html", "p1.html", "p2.html", "style.html", "email.html", "system.html", "process.html"];
```

For this problem:

- Define a form with two text fields and a button (see example below for format information).
- The first text field represents the first entry (position, not index) from the data array we want to use to generate links.
- The second text field represents the last entry (position, not index) from the data array we want to use to generate links.
- Both text fields have a default value of 1.
- Based on the text field values provided, your code will generate an HTML table where each entry is a link. For example, if the user enters 1 for both text fields, the table will have the HTML entry `syllabus.html`.
- Your code must work for values in the **data** array that are different than the ones we have provided.
- When the form is initially loaded, the message “Valid Range: 1-<END_RANGE>” where <END_RANGE> represents the last position in the array, must be displayed. For example, for the above array <END_RANGE> is 7, but this will change based on the **data** array we use.
- If the user enters an invalid range, the program will display the message “**Invalid range, valid range:** “ followed by 1-<END_RANGE>. Assuming <END_RANGE> is 7, the following will be considered a valid range: Start Range 2; End Range 5. The following will be invalid: Start Range: 8; End Range 16.
- A **main** function will define a function named **displayTable** as the function the **displayTable** button will call when selected. Feel free to add any other functionality to the main function you understand is needed.
- Notice that the HTML and JavaScript appears in a single file.
- Feel free to add any functions in addition to the displayTable and main functions.
- You can assume the above **data** array has been defined as a global variable in your code (you do not need to define it).

Form

Links Generator

Start Range: End Range:

Valid Range: 1-7

After Providing Invalid Range and Clicking on the Button

Links Generator

Start Range: End Range:

Invalid range, valid range: 1-7

After Providing a Valid Range and Clicking on the Button

Links Generator

Start Range: End Range:

p1.html
p2.html
style.html
email.html

One Possible Answer

```
<h2>Links Generator</h2>
Start Range: <input id="startNumber" type="text" value="1" size="3">
End Range: <input id="endNumber" type="text" value="1" size="3"><br><br>
<input type="submit" id="process" value="displayTable"><br><br>

<div id="display"></div>
<script>
"use strict";

let data = ["syllabus.html", "p1.html", "p2.html", "style.html",
            "email.html", "system.html", "process.html"];
main();

function genLink(resource) {
    return "<a href=\"" + resource + "\">" + resource + "</a>";
}

function main() {
    let buttonInHTMLForm = document.getElementById("process")
    buttonInHTMLForm.onclick = displayTable; // DO NOT PUT ()
    let message = "<strong>Valid Range: 1-" + data.length + "</strong>";
    document.getElementById("display").innerHTML = message;
}

function displayTable() {
    let startNumber = document.getElementById("startNumber").value;
    let endNumber = document.getElementById("endNumber").value;
    let i, body;

    if ( startNumber < 1 || startNumber > endNumber || endNumber > data.length) {
        body = "<strong>Invalid range, valid range: 1-" + data.length + "</strong>";
    } else {
        body = "<table border='1'>";
        for (let i = startNumber - 1; i < endNumber; i++) {
            body += "<tr>";
            body += "<td>" + genLink(data[i]) + "</td>";
            body += "</tr>";
        }
        body += "</table>";
    }
    document.getElementById("display").innerHTML = body;
}
</script>
```