Name: NAVIN PAUDEL StudentID: 611010

82.75

Midterm Exam

CS472 WAP July 2020 – Prof. Zijlstra

Theory Questions

a. [3 pts] Write a relative URL that would take you from http://mumstudents.org/~mzijlstra/test/index.html to: http://mumstudents.org/cs472/params.php

that's absolute

0

click here.

b. [3 pts] Describe the difference between `display: none;` and `visibility: hidden;`

→ visibility: hidden means it will not show the content (i.e text or image whatever we are using for)

→ Display: none; means basically by default for block element there will be Display: block by default but display: none means nothing nor block also.

3

c. [3 pts] What is the purpose of the `<label>` tag? (Hint: unlike desktop GUIs like Swing and JavaFX you do not need labels to put text on the screen)

→ <label> tag is just used for ~~be~~ decorating text or anything what we need in content by using ~~se~~ this label tag as selector in css

2 X click

d. [3 pts] Describe the difference between null and undefined in JavaScript

→ undefined :- In javascript, if initialization of variable or function is not found then it will declare or assign as undefined.

→ null :- X

1.5

e. [3 pts] Does JavaScript support function overloading? Explain your answer:

→ No, javascript doesnot support function overloading. If we have same function name with different variables or same variables then javascript will take later function only and ~~then~~ if we use variable not in later function then it will display as undefined.

3

f. [3 pts] Write a regular expression to match Iowa License Plates, these can be either 3 uppercase letters a space and then 3 numbers, or 3 numbers a space and then 3 uppercase letters

→ /([A-Z]{3} [0-9]{3}|[0-9]{3} [A-Z]{3})/

1.5 1S

g. [3 pts] Explain what a closure is, and how you can make one in JavaScript:

→ Closure are a functions with preserved data. It means inner function can access a ~~and~~ variable of outer function. ~~both function~~

→ we can make closure in javascript by creating outer function and inner function1 eg:

```
function(radius){
  let area = function(){
    area = radius * radius * math.PI;
  }
}
```

3

Name:_____ StudentID:_____

h. [3 pts] Explain what function currying is:

function carrying means function can be send as a parameter, function can be return, function can be assigned as a variable. ✗

0

i. [3 pts] Explain what an IIFE is and how it relates to the module pattern:

→ IIFE means Self invoked function. In module pattern we can create private functions or variables and public functions or variables and we will return public functions or variables by invoking a function by itself i.e IIFE.

3

j. [3 pts] Why is it considered a bad practice to write functions definitions inside a constructor function? (Hint: you're supposed to put them on the Contstructor.prototype instead).

→ It is bad practice because all the objects created using constructor function will have lot of burden to carry those function definitions. Rather we keep those function definition somewhere and call those functions whenever required by prototype.

3

Code Questions

1. [10] what are the colors

```
<body>
  <div id="first">First
    <ul>
      <li class="this that">
        <strong>Second</strong>
        Third
      </li>
      <li class="such">
        Fourth
        <em id="so">Fifth</em>
      </li>
    </ul>
  </div>
</body>
```

```
body { background-color: ivory; }
#first { color: blue; }
#such { color: yellow; } ✗
#so, .that { background-color: lightblue;}
ul, .that { background-color: white; }
li.this.that { color: red; } ✗
.that strong { background-color: pink; }
.such > em { color: purple; } ✗
div { color: green; }
```

10

	Foreground	Background
First	blue ✓	ivory ✓
Second	red ✓	pink ✓
Third	red ✓	white ✓
Fourth	blue ✓	white ✓
Fifth	purple ✓	light blue ✓

Excellent!

Name: NAVIN PAUDEL StudentID: 611010

2. [10] What is the output of the following JS (hint: pay attention to hoisting , closures, and this):

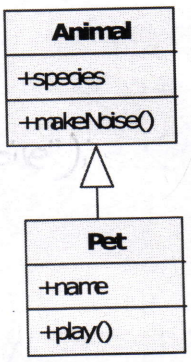
```
var x = 5;
var y = 0;
function a(n) {
    var b = function() {
        console.log(this);
        return n;
    };
    b();
    console.log(y);
    if (n > 0) {
        x--;
        n *= 2;
    }
    var y = 8;
    console.log("x:" + x + " y: " + y);
    return b;
}
var f = a(x);
console.log(f.call({'z':100}));
```

Console.log(this) → window
console.log(y) → undefined
Console.log("x:"+x+"y:"+y) → x:4 y:8
Console.log(this) → {'z':100}
Console.log(f.call({'z':100})) → 100

9.5

3. [10] Implement the following inheritance class diagram with JavaScript constructor functions. The method `makeNoise()` should console.log: "The "+species + "makes a noise" The method `play()` should console.log: name + " the "+species+" is playing"

9

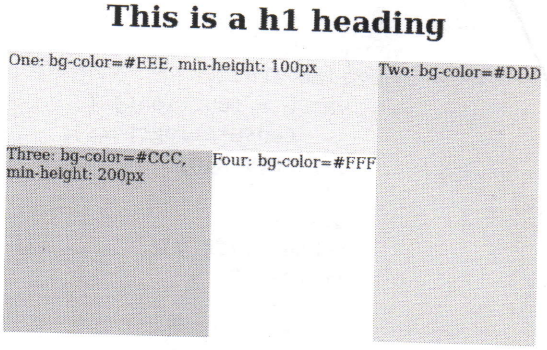



```
let Animal = function (x) {
this.species = "pet";
};
Animal.prototype.makeNoise = function () {
  Console.log("The" + species + "makes a noise");
};
function Pet() {
   this.name = "dog";
} this.species = y;
Pet.prototype.__proto__ = Animal.prototype;
Pet.prototype.play = function() {
  console.log(name + "the" + species + "is playing");
};
```

Name:_____ StudentID:_____

4. [15] Write HTML and CSS to make the following page
   - Your HTML has to be a completely valid HTML5 page (include every html tag needed, starting from scratch)
   - Write a separate "style.css" file for css
   - The resulting layout has to match the screenshot

14.75




```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <link href="style.css" rel="stylesheet">
  <title>Ques 4</title>
</head>
<body><h1>This is a h1 heading</h1>
  <div id="container">
    <div class="item1">one</div>
    <div class="item2">two</div>
    <div class="item3">Three</div>
    <div class="item4">Four</div>
  </div>
</body>
</html>
```

style.css

```
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

#container {
  display: grid;
  grid-template-areas:
  'one one two'
  'three Four two';
}

h1 { text-align }

.item1 {
  grid-area: one;
  background-color: #EEE;
  min-height: 100px;
}

.item2 {
  grid-area: two;
  background-color: #DDD;
  min-height: 300px;
}

.item3 {
  grid-area: three;
  background-color: #CCC
  min-height: 200px;
}

.item4 {
  grid-area: four;
  background-color: #FFF;
  min-height: 200px;
}
```

Name: NAVIN PAUDEL StudentID: 611010

5. [10] Write an ASCII based spinner (an animation you show during loading), using JavaScript timers based on the following HTML and CSS.

6.5

Your JS should use the revealing module pattern where your module exposes two functions: show() and hide(). Show makes it so that #loading is seen, and changes #spinner every 250 milliseconds, changing the character from – to \ to | to / and back to – . hide() makes it so that #loading is not seen, and stops the animation.

HTML	CSS

...
 	
 Loading
 -
 	
</body>
</html>
	#loading{
	display: none;
	background-color: white; opacity: 0.75;
	text-align: center;
	position: fixed;
	top: 0px; right: 0px; bottom: 0px; left: 0px;
	}
	#loading h2 { font-size: 40pt; margin: 250px 0px 0px; }
	#spinner { font-size: 60pt; }

```
let Animation = (function() {
    let clear = 0;
    function show() {
        clear = setInterval(load, 250);
    }
    function hide() {
        clearInterval(clear);
        document.getElementById("loading").style.visibility = "hidden";
    }
    function load() {
        document.getElementById("loading").style.visibility = "show
        let size = parseInt(document.getElementById("spinner").style.fontSize);
        if(size == 60)
            size += 10;
        document.getElementById("spinner").style.fontSize = size+'pt';
        if( size == 120)
            size -= 10;
        document.getElementById("spinner").style.fontSize = size+'pt'
    }
    return {
        show: show,
        hide: hide
    };
})();
```

? logic of decrease is not working my guess are all ok.

#1 look above right side.

Name:_____ StudentID:_____

6. [10] Write a JavaScript function that can take any amount of numbers, and uses functional programming techniques (no loops!) to:

- Only keep numbers that are either a multiple of 3 or a multiple of 5
- Change the numbers to strings:
  - If a number is a multiple of both 3 and 5 it should be changed to the string "FizzBuzz"
  - If a number is only a multiple of 3 it should be changed to the String "Fizz"
  - If a number is only a multiple of 5 it should be changed to the String "Buzz"
- Concatenate all the Strings into a single (space separated) String
- Return that single string.

8

```
function numberArray(...numbers){
    let arr1 = [];
arr1 = numbers.filter(num => (num%3==0)&&(num%5==0));

let arr2 = arr1.map((num)=>{
            if(num%3==0)
                num="Fizz";
            if(num%5==0)
                num="Buzz";
            if((num%3==0)&&(num%5==0))
                num="FizzBuzz";
        });return num;

let string = arr1.toString();
return string;
}
```

← without return there is no mapping

kind of works but has extra [and]