CSC 309H1 F 2019 Midterm Test
Duration — 50 minutes
Aids allowed: none

Student Number: 1 0 0 3 9 2 0 2 1 5

UTORid: h e t i a n j i

Last Name: He **First Name:** Tianjiao

Team Number: 16

Lecture Section (circle one): L0101 (MW 3), [✓ L0201 (MW 4)]

Instructor: Mark Kazakevich

Do **not** *turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm is double-sided, and consists of 4 questions and a list of some JavaScript functions/methods. *When you receive the signal to start, please make sure that your copy is complete.*

- Assume `'use strict';` for all JavaScript.
- Comments are not required except where indicated, although they may help us mark your answers.
- No error checking is required: assume all user input and all argument values are valid, unless otherwise specified.
- If you use any space for rough work, indicate clearly what you want marked.
- **Do not remove any pages from the test booklet.**

1: 5 / 5

2: 3 / 4

3: 2.5 / 5

4: 6 /10

TOTAL: 16.5 /24

Total Pages = 10

Question 1. 5 [5 marks]

20-

Indicate whether each statement is True or False by circling the appropriate answer.

TRUE [✓ FALSE] The Internet Protocol uses the three-way handshake to maintain a connection between two hosts.

TCP

[✓ TRUE] FALSE `use strict`; will sometimes (but not always) cause the following line of code to fail:
`a = 7;`

[✓ TRUE] FALSE Object constructor functions are of the same type in JavaScript as other functions.

TRUE [✓ FALSE] Anonymous functions must be called immediately where they are defined (i.e., immediately invoked).

call back

TRUE [✓ FALSE] The Transport layer is responsible for managing packets as they travel between network nodes.

Link

Question 2. [4 marks] 3

The following three lines of JavaScript are run, creating three variables in the global scope:

```
let a = 4;
let b = 1;     4      1
var s = { a: 1, b: 4, c: 2 };
```

The code fragments below are each run directly after the above three lines. They are run **independently** of each other. Beside each code fragment in the table below, write the console output when the code fragment is executed after the above three lines. If the code would cause an error, write ERROR and give a brief explanation.

Code	Output or Cause of Error	

(function () {
    if (a > 2) { let b = 3; a = 0; }   block
    console.log(a + b);
})();   0 1
	1	1

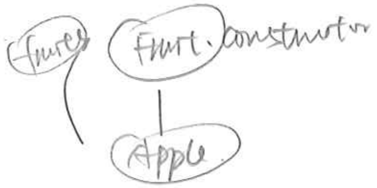
var c = 5;
for (const i = 0; i < 3; i++) {
    if (c < 4) {
        console.log(c);
    }
    var c = 3;
}
	error.
	can't reassign value to
	const type.
		1

(function(o) {   s
    o.a = 4;   s.a=4
fune scope → var s = o;
    s.b = 1;
})(s); ⇒ para
console.log(s.a); console.log(s.b);
	4
	1
		1

function foo() {
D   a(a); → a=7. a=2
    function a(b) {
        a = 7; b = 2;
    }
    console.log(a);
}
foo(); console.log(a);
	2.
	2.
		0

Question 3. [5 marks] 2.5

Consider the code below. Fill in the boxes such that the output of the `console.log` statements at the end of the code is specified by the comments beside them.
The boxes cannot contain object literals (i.e., objects surrounded by curly brackets {}).



object

```
const Fruit = {
    constructor: function(name, season) {
        this.name = name;
        this.season = season;
    }
}
```

func

```
function Apple(p) {
    Fruit.constructor .bind( Fruit ) ( p );
}

function fruits() {
    this.type = 'Fruit'
}
```

object -0.5 an object of prototype

```
Apple.prototype = Object.create( fruits );

const obj = new Apple ({ name: 'strawberry', season: 'summer' });

console.log(obj.name);   // 'strawberry'
console.log(obj.season);   // 'summer'
console.log(obj.type);   // 'Fruit'
```

this.name
this.se — sum

Question 4. [10 marks] 6

In this question, you will implement a small web page based on some user requirements. You will write some HTML and CSS, a JavaScript library, and a JavaScript file that will use the library you wrote to perform the requirements of the site.
The following are the requirements for the web page:

15X15 blue

- The page has two buttons: `Add Square` and `Add Circle`.
- The two buttons will add square or circle shapes to the DOM when clicked.
- Squares should be blue and 15 by 15 pixels, and should have the text `SQUARE` somewhere inside of them. Circles should green, with a diameter of 10 pixels, and should have the text `CIRCLE` somewhere inside of them. 10
- All shapes should be 10 pixels away from other shapes in any direction.
- Any time you add a square, it should be added on a new line (anywhere under the most recently added shape). 1
- Circles added after a square should be added on a new line (anywhere under the square).
  Any time you add two or more successive circles, each one should be added right beside (to the right of) the most recently added circle. 2
- The shapes should not be added to the `<body>` element, and should not be put in the same HTML element as the buttons.

Part (a) [2 marks] 1

Complete the HTML body below for the above requirements. It should have all elements necessary to start adding shapes, as well as **one square** and **one circle** already on the page.
You should provide proper classes and/or ids as needed. You will use these in the next parts to style and create user interactions. **Do not** write **any** JavaScript in this box.

```
<!DOCTYPE html><html>
    <head><link rel="stylesheet" type="text/css" href="shapes.css"></head>
    <body>

    <div id='buttons'>
        <button. class="square"> Add Square </button>
        <button class="Circle"> Add Circle </button>
    </div>
    <div id="dsquare">. SQUARE </div>
    <div id="dcircle"> CIRCLE </div>

     <script type="text/javascript" src="ShapeMaker.js"></script>
     <script type="text/javascript" src="shapes.js"></script>
    </body>
</html>
```

-1

Should not directly be under body -0.5

Should be class not id for the shape -0.5

Part (b) [3 marks] |

Complete the file `shapes.css` to define the styling for the shapes per the requirements. Make sure the selectors match what you put in the HTML.

shapes.css

-2

```
#dSquare{
    color : blue;
    height : 15px;
    width : 15px;
    margin : 10px;

}


#dCircle{
    color : green;
    border : 10 px;
    border-radius : 50%
    margin : 10px;
    float : left.

}
```

```
~~# text1 {~~
    ~~position~~ : absolute.

}


~~# text2 {~~
    ~~position~~ : absolute.

}.
```

Margin should be 5px -0.5
circle should be inline-block -0.5
background-color not color -1

Continued..

Part (c) [3 marks] 2 [illegible]

Complete the JavaScript library `ShapeMaker.js` below. It should have functionality for adding squares and circles to the proper place in the DOM. Make sure you are considering your HTML and CSS files when writing your code.

All DOM elements and text nodes must be created dynamically. You may not use .innerHTML, .innerText, or anything similar. Doing so will result in a 0 for this part.

—1

Sha

```
ShapeMaker.js
---------------------------------------
function ShapeMaker () {
    // empty function body.
}

ShapeMaker.prototype = {
        addSquare: function() {
        const div= document. createElement ("div")
         div.id = "dsquare".
         const body = document. queryselector ("body").
          body. appendChild ( document. createTextNode ("SQUARE"));


        },

        addCircle: function() {
         const div = document. createElement ("div");
          div.id ='dcircle".
          Const. 'body= document. querySelecter ("body").
          body. appendChild (document. createTextNode ("CIRCLE");

                                                     Should not append to
                                                     body -0.5
        }                                      Only appending textnode -0.5
}
```

Continued..

Part (d) [2 marks] 2

Now, write the code for `shapes.js`, which should set up the user interactions for the buttons when they are clicked. You should use the `ShapeMaker` library and call the appropriate functions at the appropriate times. All interactions should be set up in this file - no JavaScript should be in your HTML.

```
shapes.js
----------------------------------

const buttons= document.querySelector(#buttons);

buttons.AddEventListener("click", f1);

function f1(e){
    const shapeMaker = new ShapeMaker();
    If(e.target.ClassList.contains("square")){

        shapeMaker.addSquare(.);

    };
    If (e.target.ClassList.contains("circle")){

        Shapemaker.addCircle();

    }
}
```

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

Last Name: He First Name: Tianjiao

HTML and CSS

```
<body>, <div>, <h1>, <h2>, <p>, <ul>, <ol>, <li>,
<span>, <strong>, <em>
<img src=''>
<form>
<input type=''>
<button>
```

```
height, width, position, display, padding,
margin, border, border-radius,
color, background-color, font-family, font-size
```

JavaScript DOM functions/methods

```
document.getElementById(id)
document.getElementsByClassName(class)
document.querySelector(selector)
document.querySelectorAll(selector)

document.createElement(string)
document.createTextNode(text)
element.appendChild(element)

element.setAttribute(attributeName, value)

element.addEventListener(event, function)
event.preventDefault()

Properties:
element.className
element.id
element.classList
element.value
element.parent
element.children

array.push(object)
parseInt(string)
```

END OF TEST