CS303 Exam 2

October 7, 2020

Name:

The exam takes 2.5 hours.

Please read the exam policy before you start the exam.

Exam Policy:

You can bring one sheet of handwritten notes to the exam. You must turn this sheet and any scratch paper you use in at the end of the exam.

There is a no tolerance policy for dishonesty on exams. **You will be asked to leave the exam room immediately without a warning** for any honesty violations, which mean you will get an **NC**.

Answers should be written with a pen or pencil. If you use a pencil please bring your own eraser and sharpener. You are not allowed to borrow from other students during the exam.

All mobile phones should be turned off and stored with your coat or backpack.

You are not allowed to go to the restroom or go out of the room for water after you start the exam.

You are not allowed to ask or get extra papers from other students.

Please write down your answer clearly. If I cannot read your answer, you will not get credit.

Good luck!

1. [3] Write any additional code so the console.log gives the shown outputs

```
function foo(bar, abc, xyz) {
  if (bar < 100) return abc (bar)  //less than 100 then result is double the value
  else return xyz (bar);  //greater or equal to 100 then result is triple the value
}
console.log(foo ( 100, apple, banana)); //300
console.log(foo ( 90, apple, banana)); //180
```

CIRCLE T or F

2. T F Function declarations can be used before or after they appear in the file.

3. T F Function expressions can be used before or after they appear in the file.

4. [2] Arrow functions are best characterized as (choose best, only 1)
   a. Higher order function
   b. Function declaration
   c. Function expression
   d. Local variable
   e. Global variable

5. [1] What JavaScript command or keyword allows the developer to insert a break point in their code? ____________________

6. [3] What will be logged when the following code runs?

   a. Immediately? ____________________________________

   b. After 2 seconds? ____________________________________

   c. After 2000 seconds? ____________________________________

```
function sayHi() {
  console.log('Hello');
  return function(){console.log('Bye')};
}
setTimeout(sayHi(), 2000);
```

7. [3] Consider the following code

```
"use strict";
function perimeter(){
   console.log(this);
   return 4 * this.side;}
const shape = {side: 5,  perimeter: perimeter};
shape.perimeter();
```

What will appear in the console log?

```
const myPerim = shape.perimeter;
myPerim();
```

What will appear in the console log?

Now suppose that the “use strict” line is commented out.

```
myPerim();
```

What will appear in the console log?

```
let john = { name: "John", surname: "Smith", age: 10 };
let pete = { name: "Pete", surname: "Hunt", age: 20 };
let people = [ john, pete];
```

8. [5] Use the map function to map the people array to the following:

```
[{ fullName: "John Smith", id: 1 },
 { fullName: "Pete Hunt", id: 2 },]
```

9. [5] Retirm the smallest number of the following array using the Array.reduce method
   const numArray = [5, 44, 1, 33];

10. [5] Return the age of the youngest person from the people array above (question 8) using reduce.

11. [5] Write your own version of Array.find. Write a function, myFind that takes 2 arguments, an array and a function to apply to the array. It should work like Array.find. I.e., the input (callback) function returns true or false for each element in the original array. It should return the first element of the array for which the callback function returns true. It should not change the input array. For example,

    ```
    const numArray = [5, 11, 1, 33];
    console.log(myFind(numArray, element => element > 10))  //11
    ```

12. [5] Write a recursive function, sumToN(num) to find the sum of all integers from 1 to num

13. [7] Write a recursive function, classAdder(node), that will recurse through a tree such as below and add a property called "class" with value "classNode" to each element in the tree. For example, node3 will become { name: "p", children: null, class: "classNode" }, and similarly for the other nodes.

```
let node4 = {    name: "label",    children: null};
let node5 = {    name: "input",    children: null};
let node3 = {    name: "p",    children: null};
let node2 = {    name: "div",    children: [node4, node5]};
let node1 = {    name: "body",    children: [node2, node3],};
```

14. [3] Write nodes 1, 2, and 3 from the above problem as a singly linked list. Name the connecting property “next” instead of children. Include the name property of each node. 1 links to 2 links to 3

15. [3] Make the necessary changes to insert node5 between node1 and node2 in your linked list.

```
function makeCounter() {
  let count = 0;
  return function() { return count++; };}
```

16. [7] Recall the makeCounter function. Write a similar function with a closure, makeAccount() to keep track of an account balance.

const account1 = makeAccount();
account1(‘add’, 10); //10
account1(‘add’, 10); //20
account1(‘debit’, 4); //16

17. [3] Use the Math.min function and spread operator to find the smallest number in this array,
    const numArray = [5, 44, 1, 33];

18. [7] Write a function tenClock() that will show the time for 10 seconds. We will want this 10 second timer to display on a web page eventually, but for now log the current time to the console every second. It should show the current time in hh:mm:ss format. Recall the Date object and methods getHours, getMinutes, getSeconds.