Name:	Id:

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 no tolerance policy for academic dishonesty on exams. **You will be asked to leave the exam room immediately without a warning** if you do the following things which mean you'll get an **NC**.

1. You are caught cheating or trying to cheat.
2. Answers should be written with a Pen or Pencil, but if you want to use a pencil please bring your own eraser and sharpener. You're not allowed to borrow from other students during the exam.
3. All mobile phones should be turned off and stored with your coat or backpack.
4. You're not allowed to go to the restroom or go out of the room for water.
5. You are not allowed to ask or get extra papers from other students.

Please write down your answer clearly. If we cannot read your answer, you'll not get credit.

Good luck!

1. [5] Fill in the blanks

```
let brown= 10;
function someFun(blue) {
    let green= 108;
    if (blue >= brown) {
        blue = Math.sqrt(blue)
    }
    brown= green + blue;
    console.log(blue + " " + green + " " +  brown); _____________
    return blue;
}
console.log(brown); _____________________
console.log(someFun(36)); _____________________
console.log(brown); _____________________
console.log(green); _____________________
```

2. [2] What is the output of the code ?

```
let myObj = {
          user1: {
               fname : 'john',
               lname : 'doe'
            },
          user2: {
               fname : 'john',
               lname : 'doe'
            },
          equals: function() {
               return this.user1 === this.user2;
          }
}
alert(myObj.equals()); ____________________________________
```

3. [2] Fill in the blank for the output of the following code.

```
let anotherObject  = {color:  'green'};
let color='red';
let myObject={
     anotherObject :{color: 'blue'},
     printColor:function(){
          alert(this.anotherObject.color);
     }
}
myObject.printColor();  _____________________
```

4. [5] Write a JavaScript function "checkExam" that returns the grade number. The function receives two arguments, two arrays:

- The first input array contains the correct answers to an exam
- The second input array is "answers" array and contains student's answers.
- Return the grade number for the array of answers, giving +4 for each correct answer,-1 for each incorrect answer
-If the score < 0, return 0.

Example:
checkExam(["a", "a", "b", "b"], ["a", "c", "b", "d"]) →6
checkExam(["a", "a", "c", "b"], ["a", "a", "b", ""]) →7
checkExam(["a", "a", "b", "c"], ["a", "a", "b", "c"]) →16
checkExam(["b", "c", "b", "a"], ["", "a", "a", "c"]) →0

5. [5] Write a function ‘thisProgramIsTheBest’ which takes 3 parameters. First two parameters are objects that have property ‘age and the last one is a function, “cbFun”. If the age properties are equal, then return “Same Age!”, otherwise call the cbFun function with the input parameters as arguments, and then return “Different Ages”.

6. [6] a. Write a function, positive, that filter will use to return an array containing only positive numbers. For example, in the code below returnVal should be [10, 20, 30]. Also, fill in the blank for what the second log will show as the value of the array:

```
let array = [-10, 10, 20, -20, -10, 30]
let returnVal= array.filter(positive);
console.log(returnVal);  ____________________
console.log(array);  _____________
```

b. Write a second function, negative, that filter can use to return an array of only negative numbers

let returnVal2= array.filter(negative); //[-10, -20, -10]

7. [4] Fill in the blanks below ?

```
function haveFun(){
 let fun ={
     myFuns : []
}
let i =0;
while(i<3){
     let  myFunc = function(){
          console.log('Fun : ' + i);
     }
     fun.myFuns.push(myFunc);
     i++;
}
return fun;
}
let anotherFun = haveFun;
let result = anotherFun();
let output1 = result.myFuns[1];
let output2 = result.myFuns[0]();

console.log(anotherFun); _____

console.log(result);    _____

console.log(output1 );   _____

console.log(output2);    _____
```

8. [2] Fill in the blank

```
function fn(a,b){
   alert(a+b);
}
function fn(a){
   alert(a);
}
fn('CS');
function fn(){
   alert('Hello');
}
```

What will the alert show? _____

9. [5] Write an iterative function, sumOfSquares(n) that will compute the sum of squares of all integers from 1 to n. E.g., sumOfSquares(3) will compute 1 + 4 + 9 and return 14.

10. [5] Write a recursive version of the same function, sumOfSquares.

11. [5] The following is a node structure for a linked list. Write a recursive function, printByCredit, that logs the names of any courses with credits greater than a specified number. For example printByCredit(theList, 2) will print out the names of all courses with credits greater than 2.

```
{courseName : 'CS303',
credit : 4,
next : null }
```

function printByCredit(node, credits) {//complete the function

12. [5] Write a recursive function, listModify, which takes 2 parameters. First one is the node, second one is the modification function. Modify all the values of LinkedList with modification function. Assume the following node structure:

```
{city : 'Fairfield,
state : 'IA',
next : null}
```

function listModify(node, modifier){//COMPLETE THIS

13. [5] Write a recursive function ‘changeValues’ that takes 3 parameters. The first parameter is the node, second is the condition function and third is the modification function. Modify all the nodes that meet the condition function with modification function. Assume the following node structure:

```
{
name: “John”,
age : 5,
zipCode : 52556,
children :[]
}
```

function changeValues(node, condition, modifier) {//COMPLETE THIS

14. [10] Write code for your own version of map.
    a. Your function must be a pure function.
    b. You cannot use JavaScript's map method.
    c. The name will be myMap.
    d. It will have two parameters. The first will be an array of numbers. The second will be the map function.
    e. Write two examples of map functions that can be used with myMap, add10(number) and mult(10). add10 will add 10 to a number. mult10 will multiply a number by 10.
    f. Show how you would use myMap and add10 and mult10 with a sample array, [4, 5, 6] .

15. [10] Recall the makeCounter code:

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

Write a function, makeContact, which returns a function that keeps track of contacts. For example,
const myContacts = makeContacts();

a. makeContact should return a closure that has a private variable to keep an array of contact objects.
b. Each object in the array of contacts will have a name property and a phone property.
c. When you call the closure, myContacts, with the name of a contact it will prompt the user for a phone number and store a new contact object with that name and number.
d. If the contact name is already in the array, then it will return the phone number of that contact rather than creating and storing a duplicate contact name.