**Day 1 Lab Assignments**

1. Alert the sum of 2 numbers, and pass the sum as an argument to alert function (slef-invoking function).
2. Try arrow function:
3. With Array.filter() function, to return the odd numbers from an array.
4. With array.forech() to print the even values.
5. With array.map() to print the square of each element.
6. “An arrow function does not create its own this context, unlike the normal literal function.” – Explain with demo.
7. Try spread operator with any array of your implementation.
8. Create a student literal object that contains: name, University, faculty, and final grade. print student data in the console using template literals in this format:

{Std\_name} is a student in faculty of {fac\_name} in university {Uni\_name}

And his final grad is {grad}.

1. Make a set that holds student names.
2. Try to add repeated names, will the set accept it?
3. Print the set values using spread operator and using for…of.
4. Make a page that displays a tip for user every 3 seconds, as the following:
5. Create a generator that has an array of 10 tips, and loops on the array and each time returns the next tip.
6. Make a button that loop on the generator and display all tips [Using for…of]
7. Make another button that uses setInterval (with arrow function) to display a tip every 3 seconds from the generator.[use next()].
8. Search for at least 3 new features in ES.next (not explained in the lecture) and explain them using a demo

**Bonus:**

1. Make a Map and use the key as the day name (Saturday, Sunday,…..), and the values is a Tip of each day, and on page load make a function that loops on gets the day name of the current day, and loops on the generator to (use: for … of) to get the tip of the current day (for example: on Saturday displays the Saturday’s tip, and so on), and alert it.
2. Search for other new ESS6, 7, 8 features that weren’t explained in the lecture.

(Useful resources: <https://www.cronj.com/blog/javascript-es7-es8-new-features>)

1. Create new custom object Queue, its constructor takes maxSize, and has InQueue, DeQueue.
   1. You’ll need to define local variable toq (private) to define the last place used in the queue, and another local variable as array that will hold the elements.
   2. Create privileged method (getCurrentSize) that returns the current size of the queue.
   3. Create internal function “getQueueElements” that prints queue elements, and returns the size of queue, and create public function “viewQueue” to call it.
   4. Create new public method “returnQueue” that returns “getQueueElements” inner function as function expression (literal). The inner function should return an array of elements.
   5. Create an object from queue, and use inQueue and dequeuer functions to add some elements
   6. Then use returnQueue() function that will return a reference to function, and put it in a variable. Then call that variable (which refers to function), and call it, to return the array of elements.
      * 1. Can the inner literal function access the array local variable of the outer function? Why?
        2. What’s closure?
   7. Use prototype to add new function to queue “isEmpty()” that return Boolean.
2. Implement linked list using ES6 [create class node, and Class linkedList, to simulate linked list].

**<Script>document.write(“Thank YOU”) </Script>**