**NAME:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ DATE:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Note:*** This assignment builds understanding of following topics in JavaScript:

1. Variable hositing
2. Expressions
3. Looping
4. Functions
5. IIFE
6. Passing functions to other functions
7. The arguments object
8. Callback functions
9. Declare a variable but do not initialize it. Print its value. Now assign it a value. Print its value.
10. Print the value of a variable (before the line of declaration). Now declare and assign it a value. Print its value. Does the result look same as in Exercise 1?
11. Evaluate the value of 3 + 4 \* 5 / 6 + 7 \* 8 by hand. Verify it by logging the expression.
12. Declare an array of numbers in unsorted order (not in ascending or descending order). Write a for loop that finds the maximum value in this array. Also write a while loop that finds the sum of values in this array.
13. Write a function that accepts an array (Assume items are all numbers) and returns the average of all items in the array.
14. What does this function return when it is called??

function dummy() { }

1. Rewrite the dummy function in the above exercise using the function expression syntax.
2. Write a function that calculates and prints the sum of 2 numbers passed to it as arguments – invoke it immediately (using IIFE syntax)
3. Write a function that calculates and returns the sum of 2 numbers passed to it as arguments – invoke it immediately (using IIFE syntax). Capture the returned value (sum) in a variable.
4. Write a function that accepts any number of arguments (assume they are all numbers passed as comma-separated arguments, and not as items in an array) and returns the average of the arguments.
5. Write a function sum that calculates the sum any number of numbers passed to it. Also define another 2 functions square and cube that calculates the square and cube (respectively) of the number passed to them. Now define a final function transform that accepts 2 functions (Say f and g) followed by any number of numbers (like so)

function transform( f, g ) {

}

Any number of arguments (all numbers) may be passed to transform following f and g (the functions passed to transform).

transform( sum, square, 3, 8, 6 );

transform( sum, cube, 3, 8, 6 );

The transform function applies g on each of the number arguments and then f on the result, and returns the final result. Thus, it should return 32 + 82 + 62 and 33 +83 +63 respectively.

1. Change the previous transform function by accepting a third function h that is called and passed the final result (instead of returning it). Call it so.

transform( sum, cube, logResult, 3, 8, 6 );

logResult is a function defined somewhere in your code that shall get called by transform once the result is available.