FSW-105 Week 5 Live Meeting Summary

* Unfortunately, some plagiarism has been detected between this class and another class. Never share your GitHub link, as this will help to avoid this happening. Working with the instructor or with another group with 10-20% is fine but then finish it on your own or contact your instructor on slack to schedule more help.
* Constructor and Prototypes – creating prototypes and adding property methods to the prototype so that the common things can be extended further to any dependencies that are in place. In the next meeting you will cover the traditional object model that have properties and methods, to extend that to make a prototype and how to apply them directly. Going to give a week extension to people to turn in if they want to wait for the meeting or you can try it and submit it to see how you are understanding it on your own. If you are really wanting to be a Structural Programmer in JavaScript then this is very important.
* Array and Higher Order – Walk through on how to use advanced method of the array topic default.
* The .filter () method is going to filter out certain elements and then give you a new resulting array. Depending on the requirement the resulting array may be shrunk down to a reduced version. The goal is to go from the actual full array to the shrunk down array if it is possible.
* The .map() is to pass an array as normal, but apply the requirements so that it changes the overall state of the array. You will still get the same length array as the original.
* The .reduce() is to return the sum of an array of numbers
* The .sort() is the way to modify original array sorted either alphabetically or of numbers either high to low or low to high.
* ES6 – you do not use var everything is converted to use let and const. const is used to declare variables that will never be reassigned. Let is used to declare variables where the value could change at some point.
* I really enjoyed and feel like the conversation with Logan about how “carrot” needed to be in the square brackets in the parentheses to avoid the repetitive calling of the “carrot” every time the other items were listed. The object is to be more efficient in the writing of the code to help keep from wasting tome and being to repetitive.
* Escape Room Simulation – using the readline-sync looking up the .question to be able to send a question to the player