**CS648  
Assignment 7**

**Part 1 - Practice with Functions (5 points)**

1. Write a function called halfNumber() that accepts one argument (a number), divide it by 2, and return the result. It should log a string like "Half of 5 is 2.5.".
2. Write a function called squareNumber() that accepts one argument (a number), square that number, and return the result. It should also log a string like "The result of squaring the number 3 is 9."
3. Write a function called percentOf() that accepts two numbers, figure out what percent the first number represents of the second number, and return the result. It should log a string like "2 is 50% of 4."
4. Write a function called findModulus() that accepts two numbers. Within the function write a statement that returns the modulus of the first and second parameters. It should log a string like "2 is the modulus of 4 and 10."
5. Create a JavaScript function that accepts a certain amount of numbers as parameters. Those numbers should be collected using a prompt and the numbers should be delimited by commas. Once the values are collected, find the sum of all of the numbers combined. You will need to use a function, loop, arguments object, and several type conversion global functions to accomplish this task.

**Part 2 - The Rock, Paper, Scissors Game (10 points)**

Rock, paper, scissors is a classic 2 player game. Each player chooses rock, paper or scissors. The possible outcomes include:

* Rock destroys scissors.
* Scissors cut paper.
* Paper covers rock.

Our code will break the game into 3 phases:

1. User makes a choice. How will we collect the user’s choice?
2. Computer makes a choice. How will we collect the computer’s choice?
3. A conditional that determines who wins.

In this lab you will be responsible for figuring out some of the logic. These next steps will attempt to guide you down the right path:

1. Begin by prompting the user for their choice.
2. Create the computer’s choice. This will be generated similarly to the coin flip that you did last week. Remember though, rather than 2 options, there will be 3 here.
3. Depending on what the numeric value of the computer’s choice is, reset it to a string value of rock, paper, or scissors instead.
4. Create a conditional that checks the user’s choice in relation to the computer’s choice. Once a winner is defined, display a message within an alert box indicating who the winner is.
5. What if the result ends in a tie? Figure out how to handle that as well.
6. What if the user enters something other than rock, paper, or scissors into the prompt? Figure out how to handle that as well.

**Part 3 - Practice with Arrays (10 points)**

1. Create a string array that contains your five favorite movies. Then, use the console to display the second movie in your array.
2. Declare an array called movies using the function constructor method. Add the length of 5 into the constructor. Then, assign one of your favorite movies to each index in the array until you have 5 total movies in your array. Then, use the console to display the first movie in your array.
3. Copy your code from step 2. Add a new movie into the 3rd position within your array. Then, use the console to display the length of the array. You should now have 6 total movies stored in the array.
4. Declare an array called movies using literal notation. Then, assign one of your favorite movies to each index in the array until you have 5 total movies in your array. Now, use the delete operator to remove the first movie in the array. Use the console to display the contents of the array.
5. Declare an array called movies using literal notation. Then, assign one of your favorite movies to each index in the array until you have 7 total movies in your array. Now, use a for loop to iterate through the array and display each movie within the console window.
6. Copy the code from step 5. Now, use a for-in loop to iterate through the array and display each movie within the console window.
7. Copy the code from step 5. Using the for-in loop to iterate through the array, display each movie within the console window in a sorted view.
8. Copy the code from step 5. Under the existing array, create a new array called leastFavMovies. Populate the array with the 3 movies that you regret watching. Display both arrays within the console window so that it’s formatted to look like this (note the spacing, you must include that too):  
     
   Movies I like:  
     
   Movie 1  
   Movie 2  
   Movie 3  
   …  
     
   Movies I regret watching:  
     
   Movie 1  
   Movie 2  
   Movie 3  
   …
9. Copy the code from step 8. Now, use the concat() method to merge the two arrays together into a single array called movies. Use the console window to display the list in reverse sorted order.
10. Copy the code from step 9. Use an array function to return just the last item in the array and display it within the console window.

**Part 4 - The Product Inventory Management System (20 points)**

In this part of the assignment you will use multi-dimensional arrays to build a small inventory management system. Here are the specifics:

* The primary array will be “inventory”.
* The nested arrays (products) will each contain elements that represent sku (number), product (string), quantity (number), and cost (floating point number).
* You should have at least 5 products in your inventory.

Similar to lab 9, your application should display a title in the console window along with a command menu that features commands to view all products, update an existing product’s inventory (quantity), and exit the program.

Rather than displaying each product in a numeric list like you did in lab 9, you should display the details for each product, ordered by sku number. Here’s an example of what the user should see if they choose the view command:

4824 Shirt (10) $15.99  
6343 Jeans (22) $39.99  
3223 Socks (36) $9.99  
2233 Hat (12) $14.99  
9382 Jacket (5) $49.99

If the user selects update, a command prompt should appear that allows the user to enter a sku number and then a second prompt that allows the user to enter a new stock quantity. The quantity should then update the product of the sku number entered. If the user selects exit, the program should be terminated.

**Part 5 - Practice with the Event Model (5 points)**

In this part of the assignment you will practice working with form objects and the DOM Event Model:

1. Add a button to a web page. Use the old school HTML attribute method to attach a click event handler to the button. That handler should call a function that displays the message “I have been clicked” out into an alert box.
2. Add a button to a web page and uniquely ID the button. Write a script that finds the button by its ID. Then use the DOM Element method to associate a function with the onclick event of the button. Again, display the message “I have been clicked” out into an alert box.
3. Add a button to a web page and uniquely ID the button. Write a script that attaches an event listener to the button and listens for the click event. When it’s heard, a user-defined function should be called and the message “I have been clicked” should be displayed out into an alert box.
4. Modify the code within step 3 so that the code that displays an alert box is written directly within a callback function as opposed to a user-defined function.
5. Modify the code in step 4 so that you can write your script within the <head> tag of the page rather than underneath the button in the markup.