Lab 5 - Working with Functions and Objects (3%)

Part 1: An Average Function (1%)

REQUIREMENTS: Using **Lab-5-1.html/js**…

1. Create a function that takes 5 parameters.
2. This function will return the average of the five numbers it receives (to one decimal place). NB: Don't worry about validating the input.
3. Test your new function by outputting to the console with the values: 5, 10, 15, 20, 25. The result should be 15.0.

Now, you will use your function to determine your program average.

4. Create five variables (one for each course in this semester) and initialize them with a grade out of 100.
5. Using your custom function, create the logic that will send a success popup message if your function returns an overall average of 70 or more.
6. Create the logic that will send a "Review required" popup message if the average is below 70. Again, don't worry about validating the input.
7. Does your logic work? Make sure to test it with known values!

Part 2: I Object! (1%)

REQUIREMENTS: Using **Lab-5-2.html/js**…

1. Create a JavaScript object named ***meObject*** to represent yourself.
2. It will have four properties (you decide).
3. Use the console to output one of those properties.
4. Create a popup that will output two of those properties concatenated together. e.g. "My name is Sean and I am a teacher."
5. Comment the above line out and turn it into a method of your object.
6. Call this method.

Part 3: Make the Bank (1%)

REQUIREMENTS: Using **Lab-5-3.html/js**…

1. Create a JavaScript object to represent a bank customer.
2. Properties are: **lastName**, **branchNumber**, **accountBalance**, & **interestRate**.
3. Methods are: **makeDeposit**, **makeWithdrawal**, & **addInterest**. addInterest should simply multiply the current balance - don't worry about compound interest.
4. Each method will return a string of text: "Thank you, your current balance is now $X.XX" with the updated balance to two decimal places.

Now that you have created your object, let's call the methods…

NOTE: The string returned by your method should be output to the console after each transaction.

5. Output the account starting balance to the console.
6. Deposit $200
7. Withdraw $75
8. Add interest to the account.

EXTRA CHALLENGE: Add another property: multipleAccounts. This will hold a Boolean value. If set to *true*, the addInterest method will add .005 to the interest rate.