**COMM644  
Assignment 4**

**Practice with JavaScript OOP Concepts (10 points)**

Within a new web page, create the following applications. Remember to comment out every application once you’ve finished with that step. Be sure to place all of your JavaScript code within an external script file:

1. Create a namespace called Animal. Then, define two sub namespaces, one called Cat and another called Dog.
2. Abandon the use of namespaces and start over. Now, create two classes, one called Cat and another called Dog. The Cat class should be created using literal syntax and the Dog class should be created using the constructor syntax.
3. Create a new instance of Cat class. Directly underneath, create a new instance of the Dog class.
4. Start over and now create a new class using literal syntax called Animal. When a new instance of the Animal class is created, the message “The Animal has been created” should be displayed in the console window.
5. Now, change the above code so that is mimics a true constructor. The class should accept an argument and that value is what should be displayed in the console window. The message should be passed into the constructor at the moment that the Animal class is instantiated.
6. Start over and now create a new class using literal syntax called Animal. Define five properties within your class including properties type, breed, color, height, and length. These properties will be set in the constructor so you’ll need to pass in arguments into the function’s constructor, set the properties, and then pass in the actual values during the object’s instantiation.
7. Use a for-in loop to loop through and display all of the properties in the Animal class. You should see a total of 5 properties displayed in a list within the console window.
8. Now, create a public method called speak. Within the speak method you will use a conditional to check the type of Animal being created. If it’s a dog, return “The <color> dog is barking!” If it’s a cat return “The <color> cat is meowing!” instead. Call that method after the Animal class has been instantiated.
9. Now, convert all of your properties to private properties. Then, create a private method called checkType(). In this method you will check to see what the type is. If it’s dog, return dog, otherwise, return cat. Then, create a public method called speak that returns the value of the checkType() method. The console window should now display “The <animal type> has made a noise!”
10. Create your own String method called findWords that inherits from the native String Object. This method should find all instances of a specific word within a provided paragraph of text. It should then alert out to the user the number of time that word was found in the paragraph.

**Does My Vehicle Need An Oil Change? (10 points)**

In this lab you will be build an application that determines whether or not your vehicle needs an oil change. If it does, it should alert the user of the fact within the console window. To complete this application, follow the steps outlined below:

1. Create an abstract base class called Vehicle that accepts four parameters including make (string), model (string), total miles (number), and the mileage for the last oil change (number).
2. Within the Vehicle class, create 6 public properties named make, model, totalMiles, drivenMiles, type, and lastOilChange. Initialize the make, model, totalMiles, and lastOilChange properties with their respective arguments. The drivenMiles property should be initialized to 0 and the type property should be initialized to an empty string.
3. Now create a public method called drive() that accepts a parameter (perhaps miles). Increment and set the public drivenMiles property to the miles argument. Return the method (return this).
4. Now create a public method called checkOil(). Within this method, check to see if the total miles minus the last oil change plus the driven miles are less than or equal to 3000. If they are, return a message to the user that they need an oil change. Otherwise, return the method (return this).
5. Now create a new sub class called Car that accepts one parameter for the amount of doors that the car has.
6. Within the new Car class, initialize a public property called doorCount and set it equal to the argument.
7. Within the Car class use a conditional to check and see if the door count is greater than 2. If it is, set the type property to Sedan. Otherwise, set it to coupe.
8. Now, make sure that the Car class inherits from the Vehicle base class. Use this opportunity to pass in the make, model, total miles, and last oil change values into the Vehicle’s constructor.
9. Then, create a new instance of the Car class and pass in the number of doors that the car has into the Car’s constructor.
10. Finally, using method chaining, call the drive() method of the car object, passing in a numeric value as a parameter. At the same time, call the checkOil() method. Display the results in the console window. You will either get a message indicating your car needs an oil change or you will get the object hierarchy.

**The Recipe Card (10 points)**

Create an object to hold information on your favorite recipe. It should have properties for title (a string), servings (a number), and ingredients (an array of strings). Your object should have a publically available method that when called, will log out the recipe within the console so that the recipe information looks like this:

Guacamole  
Serves: 4  
Ingredients:   
- 3 Avocados  
- 1 Lime  
- 1 Teaspoon Salt  
- 1/2 Cup Onion  
- 3 Tablespoons Cilantro  
- 2 Diced Tomatoes  
- 1 Teaspoon Garlic  
- 1 Pinch Ground Pepper

**The Reading List (10 points)**

Keep track of which books you read and which books you want to read! Create an array of objects, where each object describes a book and has properties for the title (a string), author (a string), and alreadyRead (a boolean indicating if you read it yet). Iterate through the array of books. For each book, log the book title and book author like so: "The Hobbit by J.R.R. Tolkien". Then, use a conditional to change the output depending on whether you read it yet or not. If you read it, log a string like 'You already read "The Hobbit" by J.R.R. Tolkien', and if not, log a string like 'You still need to read "The Lord of the Rings" by J.R.R. Tolkien.' Make sure your object contains at least 5 books.

**Fill in the Blanks (10 points)**

Fill in the blanks to make this program work.

(function() {

var person = {

// add code here

};

var car = {

// add code here

};

// print vroom vroom

car.drive();

// print 'I'm rich'

if (car.price > 100000) {

person.buy(car);

}

}());