**COMM644  
Assignment 7**

**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 two classes, one called Cat and another called Dog. Both classes should be created using constructor syntax. The Cat class should be created using a named declaration and the Dog class should be created using an anonymous declaration.
2. Create a new instance of the Cat class. Directly underneath, create a new instance of the Dog class.
3. Create a new class using constructor syntax called Animal. Create a method within the Animal class that when called, displays the message “The Animal has been created” in the console window.
4. Now, replicate the above code but this time so that the class accepts an argument and that value is what is displayed in the console window. The message should be passed into the constructor at the moment that the Animal class is instantiated.
5. Start over and now create a new class using constructor syntax called Animal. Define five properties within your class including properties type, breed, color, height, and length. These properties will be set within the object 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.
6. 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.
7. 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.
8. 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!”
9. 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. Remember, you’ll need to add your method to the String object’s prototype.

**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 Jukebox Application v2 (30 points)**

In Lab 17 you created a jukebox application using object-oriented programming techniques. The downside to this application is that it was for the most part, static. Albums were added manually in code, albums were added to the jukebox manually in code, and the albums were also played manually in code. Your job in this assignment will be to do a minimal rewrite of the application that allows the user to pick an artist/album from a drop down menu on a web page and then click a play button to play the actual album. Every time the play button is clicked, you need to keep track of which albums are being played and how often. There should be a button on the screen that reads “Show my favorite album”. When the user clicks this button, it should show the user’s favorite album based on how many times it’s been played. When building the application, keep the following in mind:

1. Store artists/albums within an array
2. Bind that array to the drop down menu on the web page
3. In code, iterate through the array and programmatically create new Album class instances based on each album within the array. You’ll also need to add these instances to the jukebox.
4. Add a play button to the web page. When the user selects an artist/album from the drop down menu and clicks the play button, the play() method should be called for that album. This will track that album as being played.
5. When the user clicks the “show my favorite album” button, the favoriteAlbum() method should be called and the user’s favorite album based on number of plays should be displayed in the web page.