**Assignment 3     Total Grade: 50   (of possible 70 points)**

**Part 1 of 1  Objects     Score: 50   (of possible 70 points)**

**Score: 50   (of possible 70 points)**

Your task is to develop a simplified virtual pet app.  At this time, we will not worry about the user interface.  We’ll focus instead on the underlying object-oriented implementation.

**You will create a pet prototype.  Use an object literal to create the pet prototype:**

**var pet = {….};**

All pets are created hungry and healthy.  All pets have two Boolean properties:

**hungry:**will be initialized to true.

**ill:**will be initialized to false.

Most pets have names.   If the pets have a name, they are referred to by that name.   If they don’t have a name, they are referred to as ‘Your Pet’.

All pets need to be fed.  When they are fed, they are no longer hungry.

You’ll need to define a **feed() method** on the pet prototype.

You’ll also define a **check() method** on the pet prototype.

To check on a pet, you basically need to check **ALL**their Boolean properties and report back to the pet owner on any property that is true.

Finally you need to define a **newDay() method** for the pet object.

When newDay() is invoked, **all the Boolean properties** of the pet object that the method was invoked on are **set to true**.

**You will also create a fish prototype** that will use the pet object as its prototype.

Use Object.create() to create the fish prototype.

fish will inherit all the properties of pet and define one additional method: **clean().**

Fish need to have their tank cleaned to stay healthy so when you invoke clean() on a fish, their ill property (initially inherited from pet) is set to False.

**Finally you will create a dog prototype** that will also inherit from pet.

Use Object.create() to create the dog  prototype.

In addition to food, dogs need to be walked.  That’s how they stay healthy.  You also need to play with them so that they are not lonely.  You’ll define two methods on the dog prototype: **walk() and play().**

The dog prototype will have one direct Boolean property **lonely**that will be initially set to false but will get set to true whenever newDay() is invoked.  Of course when you play with a dog, they are no longer lonely.

**Testing:**

Once you have created your three prototypes, you’ll be able to run the test scenario described in the file, VirtualPetTestScenario.pdf with a pet fish named Wanda and a pet dog named Fido. The file VirtualPetTestScenario.pdf is available under Resources -> Assignment 3 Files.

As illustrated in the test scenario, the various methods**return text strings** such as ‘Good morning’ and ‘Fido is full’.   More details on that are included in the template file virtualpet.js, also available under Resources.  Make sure you **read the comments provided for each method.**

The template virtualpet.js also includes the JavaScript code to run the above scenario. The expected console output is documented there. It is also included in VirtualPetTestScenario.pdf. Feel free to add your own test cases.

**How do I get started?**

**Download or copy and paste the template file virtualpet.js available under Resources.** **Then go to Komodo Edit or Scratchpad and open it from there.**You can then complete the assignment by modifying it then running it in Scratchpad.

**How do I submit my work?**

Upload your solution in the modified file virtualpet.js.  Just make sure you test it first.  Compare your console output to the expected output.

**Grading Rubric:**

Pet prototype:  40 points total

Object creation and initialization: 6 points

newDay(): 14 points

check():  14  points

feed() : 6 points

Fish Prototype: 12 points total

Object creation and initialization: 6 points

clean: 6 points

Dog Prototype: 18 points total

Object Creation and initialization: 6 points

walk(): 6 points

play(): 6 points

**Answer**

* application/javascript[virtualpet.js](https://myetudes.org/access/mneme/content/private/mneme/09ae2205-2717-4bfc-00cf-33f5bdcd7b48/submissions/14806342/35af1c6c-1c95-488a-0058-3cbb1f7a1f59/virtualpet.js)

[[https://myetudes.org/ambrosia_library/icons/collapse.gif](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14806342/list) Model Answer](https://myetudes.org/portal/tool/acd42055-9bd4-4630-8071-c0425c2388c3/review/14806342/list)

/\*

 \* Virtual Pet App - Solution

 \*

 \*/

'use strict';

// Create the pet protoptype

var pet = {

    hungry: true,

    ill: false,

    name: 'Your Pet'

};

pet.feed = function () {

    // The pet is no longer hungry.

    // Return a string indicating that the pet is full.

    this.hungry = false;

    return this.name + ' is full.';

};

pet.newDay = function () {

    // Set ALL the boolean properties to true.

    // Return 'Good morning!'

    for (var prop in this) {

        if (typeof this[prop] === 'boolean') {

            this[prop] = true;

        }

    }

    return 'Good morning!';

};

pet.check = function () {

    // Check ALL the boolean properties of the pet object.

    // Return a string of the form (for all the true properties):

    // pet name is property1. pet name is property2. etc...

    // If all the Boolean properties are False, return a string of the form:

    // pet name is fine.

    var result = '';

    for (var prop in this) {

        if (this[prop] === true) {

            result += this.name + ' is ' + prop + '. ';

        }

    }

    // if the result is still empty '', the

    // or statement below will return the second operand

    return result || this.name + ' is fine.';

};

// Create the fish prototype.

var fish = Object.create(pet);

fish.clean = function () {

    // Set the object ill property to false.

    // Return a string of the form: 'pet name likes the clean tank.'

    this.ill = false;

    return this.name + ' likes the clean tank.';

};

// Create the dog prototype.

var dog = Object.create(pet);

// initialize the lonely property

dog.lonely = false;

dog.walk = function () {

    // Set the object's ill property to false.

    // Return a string of the form: 'pet name enjoyed the walk!'

    this.ill = false;

    return this.name + ' enjoyed the walk!';

};

dog.play = function () {

    // Set the object's lonely property to false.

    // Return a string of the form:  'pet name loves you.'

    this.lonely = false;

    return this.name + ' loves you.';

};

/\*

 \* Virtual Pet App - Tests

   The expected console output for this test is:

    Fido is hungry.

    Fido is full.

    Fido is fine.

    Wanda is hungry.

    Wanda is full.

    Wanda is fine.

    Good morning!

    Good morning!

    Fido is hungry. Fido is lonely. Fido is ill.

    Fido is full.

    Fido is lonely. Fido is ill.

    Fido loves you.

    Fido is ill.

    Fido enjoyed the walk!

    Fido is fine.

    Wanda is hungry. Wanda is ill.

    Wanda is full.

    Wanda is ill.

    Wanda likes the clean tank.

    Wanda is fine.

    Fido is fine.

    Fido enjoyed the walk!

    Your Pet is hungry.

    Good morning!

    Your Pet is bored. Your Pet is hungry. Your Pet is ill.

 \*

 \*/

// Create a dog and a fish pet objects.

var myDog = Object.create(dog);

myDog.name = 'Fido';

var myFish = Object.create(fish);

myFish.name = 'Wanda'

console.log(myDog.check());      // Fido is hungry.

console.log(myDog.feed());       // Fido is full.

console.log(myDog.check());      // Fido is fine.

console.log(myFish.check());     // Wanda is hungry.

console.log(myFish.feed());      // Wanda is full.

console.log(myFish.check());     // Wanda is fine.

console.log(myDog.newDay());     // Good morning!

console.log(myFish.newDay());    // Good morning!

console.log(myDog.check());      // Fido is hungry. Fido is lonely. Fido is ill.

console.log(myDog.feed());       // Fido is full.

console.log(myDog.check());      // Fido is lonely. Fido is ill.

console.log(myDog.play());       // Fido loves you.

console.log(myDog.check());      // Fido is ill.

console.log(myDog.walk());       // Fido enjoyed the walk!

console.log(myDog.check());      // Fido is fine.

console.log(myFish.check());     // Wanda is hungry. Wanda is ill.

console.log(myFish.feed());      // Wanda is full.

console.log(myFish.check());     // Wanda is ill.

console.log(myFish.clean());     // Wanda likes the clean tank.

console.log(myFish.check());     // Wanda is fine.

console.log(myDog.check());      // Fido is fine.

console.log(myDog.walk());       // Fido enjoyed the walk!

// Create a fish with no name

var myNewFish = Object.create(fish);

console.log(myNewFish.check());  // Your Pet is hungry.

// Create a new boolean property for myNewFish

myNewFish.bored = false;

console.log(myNewFish.newDay()); // Good morning!

console.log(myNewFish.check());  // Your Pet is bored. Your Pet is hungry. Your Pet is ill.

**Comments**

You did a good job on the methods clean, walk and feed.

The play method needs to also access the object through 'this' and not the dog prototype.  Instead of:

dog.play = function () {

    dog.lonely = false;

    return this.name + " loves you.";

}

We need to write:

dog.play = function () {

    this.lonely = false;

    return this.name + " loves you.";

}

The newDay method needs to set ALL the boolean properties of the object not just hungry, ill and lonely (we'll have a bored property on the fish).

Similarly the check method needs to check on all the boolean properties.

Take a look at the model answer to see how to do that with a for... in loop as suggested in the Hints.

 1 of 1