**BCDV 1006**

**Lab assignment:**

Evaluation: Unless otherwise told, this lab must be submitted by the end of the class it is assigned in to receive 100%. This lab may be submitted by 9pm the next day, with a 35% penalty (ie, the highest possible mark will be 65%). This lab will not be accepted after that point.

**To submit this lab, upload it to blackboard.**

- Create a object literal that represents a cat. The cat should have the attributes 'name', 'age', and 'breed', such that, if the cat is stored in a variable named ''myCat”, you can access these values by typing `myCat.age` and `myCat.name`. Add any other attributes you’d like.

- Use dot access to get properties of the cat.

- Use bracket access to get properties to the cat, eg: `myCat[‘name’]`.

- Use a variable to access one of myCat’s properties, eg:

let propertyName = “breed”;

console.log(myCat[propertyName]);

* Try setting propertyName to different values

- Create an array of the cat’s attributes and iterate through the cat’s properties, eg:

var catributes = [‘name’, ‘age’, ‘breed’];

for (let idx = 0; idx < catributes.length; idx++ {

// fill out this part

}

- Create a Cat class, your Cat class should take the same arguments as your cat literal objet.

- Add a greeting method to your cat class, that uses `this` to introduce your cat. You should be able to reproduce the following:

let mittens = new Cat('Mittens!', 'Siamese', 5);

let bubba = new Cat('Bubba', 'Tabby', 7.5);

console.log(mittens.greeting());

// This should output “Meeeeeow, hellow, I'm Mittens!”

console.log(bubba.greeting());

// This should output “Meeeeeow, hellow, I'm Bubba”

- Add a “wag tail” method to the Cat prototype, which outputs the following:

console.log(mittens.wagTail());

// Should output Mittens! wags tail happily

console.log(bubba.wagTail());

// Should output Bubba wags tail happily

- You can check if an instance is an instance of a class using the `instanceof` keyword. In your console, try this:

`mittens instanceof Cat`.

**Bonus advanced questions**

Newer versions of JS introduced class based objects. Read and do the class section of the Turing FE OO lesson, “Classes and Object Instances” on

: <https://frontend.turing.io/lessons/module-1/objects-this-classes.html> .

Be careful to note the difference between Prototype inheritance and Class based inheritance. Prototype inheritance is old, more flexible, and is what Class based inheritance is built on. Some people find Class based inheritance easier to work with, however.