**Lab 13**

**Type up the code on page 577, Test drive the prototype with some dogs, to experience inheritance in JavaScript for 25 points.**

**Please check the link below for a better layout of the code:**

[**https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter13/dog.html**](https://github.com/AhmedAbdelRazak/RCC/blob/master/Chapter13/dog.html)

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8" />

        <title>Inheritance Lab 13</title>

    </head>

    <body>

        <script>

            //Dog constructor

            function Dog(name, breed, weight) {

                this.name = name;

                this.breed = breed;

                this.weight = weight;

            }

            //Here is where we add properties and methods to the dog prototype

            Dog.prototype.species = "Canine";

            //Here, we add one property and three methods to the prototype

            Dog.prototype.bark = function () {

                if (this.weight > 25) {

                    console.log(this.name + " says Woof!");

                } else {

                    console.log(this.name + " says Yip!");

                }

            };

            Dog.prototype.run = function () {

                console.log("Run!");

            };

            Dog.prototype.wag = function () {

                console.log("Wag!");

            };

            //Here we are going to create multiple dog objects like what we did in chapter 12

            var fido = new Dog("Fido", "Mixed", 38);

            var fluffy = new Dog("Fluffy", "Poodle", 30);

            var spot = new Dog("Spot", "Chihuahua", 10);

       //Based on page 578, It is required for me that the output have 'spot' says WOOF!

      //That's why the method below is required.

            spot.bark = function () {

                console.log(this.name + " says WOOF!");

            };

            fido.bark();

            fido.run();

            fido.wag();

            fluffy.bark();

            fluffy.run();

            fluffy.wag();

            spot.bark();

            spot.run();

            spot.wag();

        </script>

    </body>

</html>

Output:

Graphical user interface, text, application, email

Description automatically generated