**1.**

What is a constructor call?

A constructor function call creates an object linked to it's own prototype.

Eg. Uses key word ‘new’;

Function a() {

var b = new a();}

Eg.

function TeslaCar(name) {

this.name = name;

}

TeslaCar.prototype.logName = function () {

console.log(this.name);

};

var modelS = new TeslaCar('Model S');

var modelX = new TeslaCar('Model X');

console.log(modelS.constructor === TeslaCar);

console.log(modelS.constructor === modelX.constructor);

console.log(Object.getPrototypeOf(modelS) === TeslaCar.prototype);

console.log(Object.getPrototypeOf(modelX) === TeslaCar.prototype);

**2.**

What is a .constructor property?

It’s a property that a prototype

**3.**

* What is [[Prototype]]? (internal private link between one object and the other.
* Where does it come from? When new object is created. ObjectCreate.
* How does it affect an object?

**4.**

How do we find out where an object's [[Prototype]] points to? Describe 3 ways.

Console.log(b.\_\_proto\_\_); OLD – DO NOT USE!!!

Console.log(Object.getPrototypeOf(b));

Console.log(b.constructor.prototype);

**5.**

Draw a diagram and explain the following code:

function Car(colour) { this.colour = colour; } Car.prototype.drive = function () { console.log('Driving'); }; var redCar = new Car('red');

**6.**

Draw a diagram and explain why the following code logs 'function Tesla() {}':

function Tesla() {} console.log(Tesla.toString()); // 'function Tesla() {}'

**7.**

Draw a diagram and explain the following code:

function Car(make, model) { this.make = make; this.model = model; } Car.prototype.drive = function () { console.log('Driving'); }; function LuxuryCar(make, model, price, colour) { Car.call(this, price, colour); this.price = price; this.colour = colour; } LuxuryCar.prototype.drive = function () { console.log('Driving fast'); }; LuxuryCar.prototype = new Car(); LuxuryCar.prototype.constructor = LuxuryCar; var aventador = new LuxuryCar('Lamborghini', 'Aventador', '£260,040', 'white'); console.log(aventador.drive());

**8.**

Give a simplest example of a closure.

**9.**

Convert this Module Pattern to Prototype-based:

var Computer = (function () { function on() { console.log('On'); } function off() { console.log('Off'); } function restart() { off(); on(); } return { on: on, off: off, restart: restart }; })();