Javascript Scope Exercises

1. Determine what this Javascript code will print out (without running it):

```
x = 1;

var a = 5;

var b = 10;

var c = function (a, b, c) {

        var x = 10;

        document.write(x); // print 10

        document.write(a); // print 8

        var f = function (a, b, c) {

                b = a;

                document.write(b); // print 8

                b = c; //

                var x = 5;

        }

        f(a, b, c);

        document.write(b); // print 9

}

c(8, 9, 10);

document.write(b); // print 10

document.write(x); // print 1
```

2. What is the difference between a method and function?

ANSWER: Method is associated with an object, while a function is not. We can call function by name directly, for the method, we can call it associated with an object.

3. What does 'this' refer to when used in a Java method?

ANSWER: ‘this’ is a reference to the current object

4. What does 'this' refer to when used in a JavaScript method?

ANSWER: In a method: this refers to the owner object.

5. What does 'this' refer to when used in a JavaScript constructor function?

ANSWER: ‘this’ refer the new object in constructor

6. Assume object *x* is the prototype for object *y* in Javascript. Object *x* has a method *f*( ) containing keyword 'this'. When *f* is called by *x.f*( ), what does 'this' refer to?

ANSWER: ‘this’ refer to x

7. What is a free variable in JavaScript?

ANSWER: Free variables are the variables that are neither locally declared nor passed as parameter.

8. Create an object that has properties with name = "fred" and major="music" and a property that is a function that takes 2 numbers and returns the smallest of the two, or the square of the two if they are equal.

```
{
        'name': "fred",
        'major': "music",
        'fn': (a, b) => {
                if(a === b) {
                        return a * b;
                }
                return Math.min(a, b);
        }
}
```

9. Write Javascript code for creating three *Employee* objects using the "new" keyword and a constructor function. *Employee* objects have the following fields: name, salary, position.

```
class Employee {
        constructor(name, salary, position) {
                this.name = name;
                this.salary = salary;
                this.position = position;
        }
}
const emp1 = new Employee("Ganzorig", 130000, "Software Engineer");
const emp2 = new Employee("John", 120000, "DevOps");
const emp3 = new Employee("Doe", 110000, "SRE");
```

10. Write a Javascript function that takes any number of input arguments and returns the product of the arguments.

```
function fn(...params) {

        return { ...params }

}
```

11. Write an arrow function that returns the maximum of its three input arguments.

```
const findMax = (a, b, c) => {

        return Math.max(a, b, c);

}

console.log(findMax(1, 2, 4));
```