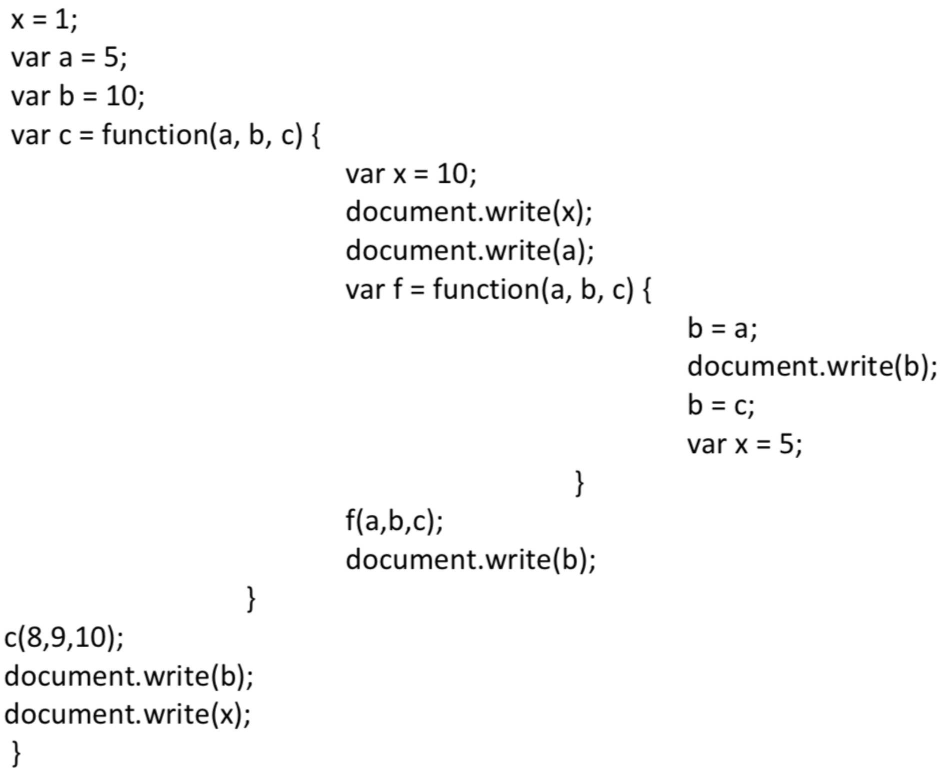
Javascript Scope Exercises

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);
                    document.write(a);
                    var f = function(a, b, c) {
                                        b = a;
                                        document.write(b);
                                        b = c;
                                        var x = 5;
                                  }
                    f(a,b,c);
                    document.write(b);
              }
c(8,9,10);
document.write(b);
document.write(x);
 }
```

10 8 8 9 10 1

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

JavaScript methods are actions that can be performed on objects. A JavaScript method is a property containing a function definition. Methods are functions stored as object properties.

In JavaScript every function is an object. An object is a collection of key:value pairs. If a value is

a primitive (integer, string, boolean), or another object, the value is considered a property. If a value is a function, it is called a 'method'.

Within the scope of an object, a function is referred to as a method of that object. It is invoked

from the object namespace 'MyObj.theMethod()'. Since we said a function is an object, a function within a function is considered a method of that function.

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

This keyword in java can be used inside the Method or constructor of Class. It(this) works as a

reference to the current Object whose Method or constructor is being invoked. This keyword can

be used to refer to any member of the current object from within an instance Method or a

constructor.

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

This keyword in JavaScript method refers the object that own the method.

Example

```
var person = {
        firstName : "John",
        lastName : "Doe",
        id : 5566,
        myFunction : function() {
                return this;
        }
};
```

In the example this refers the Person object.

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

In a constructor function this does not have a value. It is a substitute for the new object. The

value of this will become the new object when a new object is created.

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?

This refers to the prototype x.

7. What is a free variable in JavaScript?

A free variable in JavaScript is: a variable referred to by a function that is not one of its

parameters or local variables.

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.

```
function person(Fikir,music){
        this.Name=Fikir;
        this.Major=music;
        this.smallest=function(x,y){
                if(x<y){
                        return x;
                }else if(y<x){
                        return y;
                }else{
                        return x*x +" and "+ y*y;
                }
        }
}
var person1=new person("Fikir","music");
console.log("Name is :" + person1.Name );
console.log("Smallest Number is :"+person1.smallest(5,9));
console.log("The square out put is :"+person1.smallest(9,9));
```

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.

```
function Employee(name,salary,position){
        this.firstName=name;
        this.Salary=salary;
        this.Position=position;
}

Employee.prototype.information= function(){
return this.firstName+", "+this.Salary+", "+ this.Position;
}

var employee1=new Employee("Fikir",6000,"Developer");
var employee2=new Employee("Mar",7000,"Manager");

var employee3=new Employee("Emnet",2600, "Secretary");


console.log("Employee 1: "+employee1.information());

console.log("Employee 2: "+employee2.information());

console.log("Employee 3: "+employee3.information());
```

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

```
function func(...args) {
   if(args.length > 0){
       var total = args[0];
         for (i = 1; i < args.length; i++) {
             total = total * args[i];
         }
         console.log(total);
   }
}
func(1, 2, 3);
```

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