Javascript Scope Exercises

Mohammad Haroon Torab
ID- 612717

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);
}
```

Ans:
document.write(b); // 10
document.write(x); //10

2. What is the difference between a method and function?
Ans:
JavaScript Method: *is a property of an object that contains a function definition. Methods are functions stored as object property.*
JavaScript functions: *is a block of code written to perform some specific set of tasks. We can define a function using the function keyword, followed by name and optional parameters.*

3. What does 'this' refer to when used in a Java method?
Ans:
The this keyword refers to the current object in a method or constructor. The most common use of this keyword is to eliminate the confusion between class attributes and parameters with the same name(because a class attribute is shadowed by a method or constructor parameter).

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

*In **JavaScript**, the property of an object can be a **method** or a simple value. When an object's **method** is invoked, then this **refers** to the object which contains the **method** being invoked. ... So the value of this depends on how a **method** is being invoked.*

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

Ans:

*In **JavaScript**, the thing called this is the object that "owns" the code. The value of this , when used in an object, is the object itself. In a constructor function this does not have a value. It is a substitute for the new object.*

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?

Ans:

this in the above example refers to the function f().

7. What is a free variable in JavaScript?

Ans:

***Free variables** are simply 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.

Ans:
```
var newObj = {
    name: "fred",
    major: "music",
    smallestOrSquare: function(numOne, numTwo) {
        if (numOne < numTwo) {
            return this.numOne;
        } else if (numOne > numTwo) {
            return this.numTwo;
        } else {
            return Math.sqrt(this.numOne * this.numOne + this.numTwo * this.numTwo)
        }
    }
}
```

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.

Ans:

```
function Person(name, salary, position) {
  this.name = name;
  this.salary = salary;
    this.position = position;
}
const personOne = new Person('testNameOne', 'testSalaryOne', 'testPositionOne);
const personTwo = new Person('testNameTwo', 'testSalaryTwo', 'testPositionTwo);
```

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

Ans:

```
function add(){
    console.log(arguments);
    var sum = 0;
    for(var i = 0; I < arguments.length; i++){
        sum += arguments[i];
    }
    return sum;
}
console.log(add(5, 7, 3):  //15
```

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

Ans:

```
constLargestOfThree(arr) =>arr.map(el =>Math.max(…el)).

// you can even do a combination
function largestOfFour(arr) {
```

```
    return arr.map(el =>return Math.max(…el))
}
```