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);
 }
```

➔ It will write to a document with text "10 8 8 9 10 1" respectively

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

➔ **<u>Function</u>**
   Function is a code written to perform specific task. It can be invoked by calling functionName with ();

   <u>Syntax:</u>
```
function functionName(parameters) {
  // Content
}
```

   <u>Method</u>
   Method is a property of an object that contains a function definition.

   <u>Syntax:</u>
```
object = {
 methodName: function() {
   // Content
  }
};
```

3. What does 'this' refer to when used in a Java method?
   - ➔ In Java method, this refers to object of current class.

4. What does 'this' refer to when used in a JavaScript method?
   - ➔ The JavaScript this keyword refers to the object it belongs to. It has different values depending on where it is used:
     - In a method, **this** refers to the owner object.
     - Alone, **this** refers to the global object.
     - In a function, **this** refers to the global object.
     - In a function, in strict mode, **this** is undefined.
     - In an event, **this** refers to the element that received the event.
     - Methods like call(), and apply() can refer **this** to any object.

5. What does 'this' refer to when used in a JavaScript constructor function?
   - ➔ The keyword **this** inside the constructor function points to the newly created 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?
   - ➔ **this** refers to x object.

7. What is a free variable in JavaScript?
   - ➔ Free variable 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.
   - ➔

```
var obj = {
    name: "fred",
    major: "music",
    sum: function (x, y) {
      if (x == y) {
        return x * x;
      }
      return Math.min(x,y);
    }
  }
```

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;
    }
  }

  var employee1 = new Employee("Sujan",120000,"Software Engineer");
  var employee2 = new Employee("Ram",80000,"Junior Engineer");
  var employee3 = new Employee("Shyam",90000,"Mid Engineer");
```

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

➔
```
function product(...value) {
    let z = 1;
    for (let i = 0; i < value.length; i++) {
      z *= value[i];
    }
    return z;
  }

  console.log(product(2, 3, 4,5));  //output is 120
```

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

➔
```
  var max = (x, y, z) => Math.max(x, y, z);
var output = max(1,2,3);  // output is 3
```