1. print output is undefined889101

document.write(x)=undefined, because x is hoisted and given value undefined during first pass.

document.write(a) = 8, this a gets its value from function call c(8,9,10)

document.write(b) = 8, b is set to a ,which is 8 from function call f(a,b,c)

document.write(b) = 9, once function call f(a,b,c) is completed, b remains 9, which is from function call c(8,9,10)

document.write(b) = 10, here comes from the global assignment b=10

document.write(x) = 1, from global assignment x=1

2.Global scope is page/window level variables, objects and functions whereas local scope is limited to function/local level access.

3. a)No, because outer scope, A, cannot access inner scopes, B and C.

b)Yes, inner scope, B, can access any outer scope variables.

c)No, because B is in outer scope of C.

d)Yes, because C is in inner scope of A.

e)Yes, because C is also in inner scope of B

4.81, because x is defined and assigned 9 in the outer scope of myFunction . So it is accessible by this function. And 25, because x is modified to 5 before the second call to myFunction

5.10, because there is var foo=10 inside function bar, foo is hoisted and hence it is undefined. Therefore if(!foo) is true.

1. print output is undefined889101

document.write(x)=undefined, because x is hoisted and given value undefined during first pass.

document.write(a) = 8, this a gets its value from function call c(8,9,10)

document.write(b) = 8, b is set to a ,which is 8 from function call f(a,b,c)

document.write(b) = 9, once function call f(a,b,c) is compeleted, b remains 9, which is from function call c(8,9,10)

document.write(b) = 10, here comes from the global assignment b=10

document.write(x) = 1, from global assignment x=1

2.Global scope is page/window level variables,objects and functions whereas local scope is limitted to function/local level access.

3. a)No, because outer scope,A, cannot access inner scopes,B and C.

b)Yes, inner scope,B, can access any outer scope variables.

c)No,because B is in outer scope of C.

d)Yes, because C is in inner scope of A.

e)Yes, because C is also in inner scope of B

4.81, because x is defined and assigned 9 in the outer scope of myFunction . So it is accessible by this function.And

25, because x is modified to 5 before the second call to myFunction

5.10, because there is var foo=10 inside function bar, foo is hoisted and hennce it is undefined. Therefore if(!foo) is true.

6. var count=(function() {

var counter = 0;

const addcount = function add(){

count += 1;

};

const resetcount=function reset() {

count = 0;

};

return{

add:addcount,

reset:resetcount

};

})();

7. Free Variable is counter. Free variables are those that are neither locally declared nor passed as a parameter, and these are

Variables that the function takes (read and write) in the enclosing scope where is declared the closure or in a parent scope.

8. let make\_adder = function (inc) {

let counter = 0;

return function add() {

counter += inc;

}

}

9.Using IIFE Module Pattern. The module pattern helps to reduce many golobal function and variable names to fewer ones.Because we can declare and immediately call anonymous functions.

10.const employee = (function () {

let name;

let age;

let salary;

const getAge = function () {

return age;

};

const getSalary = function () {

return salary;

};

const getName = function () {

return name;

};

return {

setAge: function (newAge) {

age = newAge;

},

setSalary: function (newSalary) {

salary = newSalary;

},

setName: function (newName) {

name = newName;

},

increaseSalary: function (percentage) {

salary += (getSalary() \* percentage);

},

incrementAge: function () {

age = getAge() + 1;

},

}

})();

11. employee.address;

employee.setAddress = function (newAddress) {

address = newAddress;

};

employee.getAddress = function () {

return address;

};