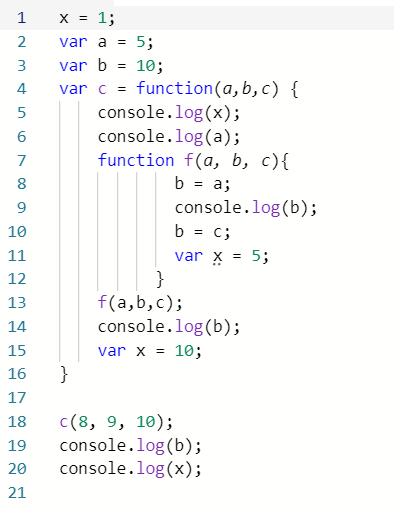
1.



**Phase 1 – hoisting:**

var a, b

var function c(a, b, c) { … }

**Phase 2– execution/evaluation of statements and expressions:**

x=1

a=5

b=10

call c(8 , 9, 10)

Next, in c() function’s lexical environment:

**Phase 1– hoisting:**

var x

function f(a, b, c) { … }

**Phase 2– execution of statements and expressions:**

a=8, b=9, c=10 //assign value a, b, c at function call

document.write(x) 🡪 undefined (x is hoisted within c())

document.write(a) 🡪 8 (a is assigned at function call)

call f(a, b, c)

Next in f() function’s lexical environment:

**Phase 1 – hoisting:**

var x

**Phase 2 execution of statements and expressions:**

a=8, b=9, c=10 //assign value a, b, c at function call

b=a //b=8

document.write(b) 🡪 8 (b is assigned to a=8 in the previous line)

b=c //b=9

x=5

//exit f()

document.write(b) 🡪 9 (b is assigned at function call)

x=10

//exit c()

document.write(b) 🡪 10 (b is assigned to 10 in the global scope)

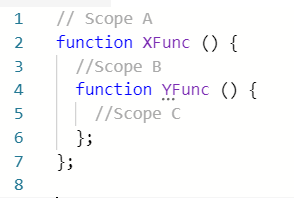
document.write(x) 🡪 1 (x is assigned to 1 in the global scope)

2.

**Global scope** means all of variables declared outside function we call and these variables have global scope.

**Local scope** means all of variables declared inside function we call and they have local scope.

3.



a) No. Because B and C are local scope.

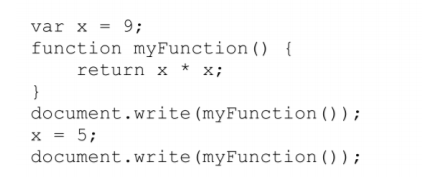
b) Yes. Because A is global scope.

c) No. C is local scope.

d) Yes. Because A is global scope.

e) Yes. By closure.

4.



**Phase 1: hoisting**

var x

var myFunction() { … }

**Phase 2: execution statements and expressions:**

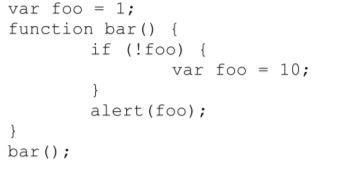
x=9

document.write(myFunction()); 🡪 81 (JS Engine goes up to global scope and finds x = 9)

x=5

document.write(myFunction()); 🡪 25 (JS Engine goes up to global scope and finds x = 5)

5.



**Phase 1: hoisting**

Var foo

Var bar = function () { … }

**Phase 2: execution statements and expressions:**

foo=1

call bar()

**Phase 1: hoisting**

var foo 🡪 JS engine goes up to global scope and finds var foo

**Phase 2: execution statements and expressions:**

if(!foo) 🡪 true because !undefined = true

foo=10

alert(foo) 🡪 10 (foo=10 in the previous if)

//exit