

question-1

```
x = 1;
var a = 5;
var b = 10;
var c = function() {
  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);
  var x = 10;
}
c(8, 9, 10);
document.write(b);
document.write(x);
```

-- -- -- -- --

First, in the Global Lexical Environment:

Phase 1 - hoisting

```
var x = undefined
var a = undefined
var b = undefined
var c = [function(){}]
```

Phase 2 - Execution/Evaluation

```
x = 1
a = 5
b = 10
calls c()
```

Next, in the c() function's Lexical Environment:

Phase 1 - hoisting

```
var x = undefined
var f = [function(){}]
```

Phase 2 - Execution/Evaluation

```
document.write(x); //x = 1 //global value
document.write(a); //a = 5 //global value
calls f(8, 9, 10)
```

Phase 1 - hoisting

```
var x = undefined
var a = undefined
var b = undefined
var c = undefined
```

Phase 2 - Execution/Evaluation

```
a = 8
b = 8 //b=a
document.write(b); //8
c = 10
x = 5 //x updated
```

document.write(b); //10

x = 10 //nested scope

```
document.write(b); //10 //global variable,
document.write(x); //1 //global variable
```

question-2

Variables defined inside a function are in local scope while variables defined outside of a function are in the global scope. Each function when invoked creates a new scope.

question-3

Given: javascript code, below

```
//scope A
function XFunc(){
  //Scope B
  function yfunc(){
    //scope C
  }
}
```

- (a) Do statetments in Scope A have access to variables defined in scope B and C?
No, scope A can't access nested scope/functions
- (b) Do statetments in Scope B have access to variables defined in scope A?
Yes, Scope of A global to scope of B.
- (c) Do statetments in Scope B have access to variables defined in scope C?
No, scope B can't access below nested scope
- (d) Do statetments in Scope C have access to variables defined in scope A?
yes, the act like global scope
- (e) Do statetments in Scope C have access to variables defined in scope B?
yes, the act like global scope

question-4

```
var x = 9;
function func(){
  return x*x;
}
console.log(func())           //output: 91  //using global variable x =9
x = 5;
console.log(func())           //output: 25  //using updated global variable x =5
```

question-5

```
var foo = 1;
function bar(){
  if(!foo){
    var foo = 10;
  }
  alert(foo);           //alert 10, since variables declared with var, have function level
  scope
}
bar();
```