**Q. 13. What will be the outputs of following questions and why?**

**a)**

var someText = 'I am awesome text';  
(function () {  
    console.log ('Original value was: ' + someText);  
    var someText = I am super awesome text;  
    console.log ('New value is: ' + someText);  
}) ();

**Output:** Uncaught SyntaxError: Unexpected identifier

**Reason :** Because var someText is a variable which is storing the string i.e. I am super awesome text. But when we initialize any string to a variable, it must have quotes. As quotes are not given to the string, we got an error i.e., Uncaught SyntaxError: Unexpected identifier.

**b)**

var a = 1;  
function b () {   
 a = 10;

return;  
   
  function a () {}  
  }  
b();  
console.log(a);

**Output:** **1**

**Reason :** Because var is a global variable in this code which can be accessed anywhere throughout the program. Though, a is again used in function b( ) and initialized with value 10, but the scope of that a is limited only for function b ( ). That a with value 10 can not be accessed out of function b( ) as it is local for function b( ). So console.log(a) has accessed global local variable with value 1. So, we got an out as 1.

**c)**

function foo () {  
    function bar () {  
        return 10;  
  } return bar ();  
    function bar () {  
        return 18;  
    }  
}  
alert(foo ());

**Output:** 18 with alert pop up message on webpage.

**Reason:** Function bar is written inside the function foo () and function bar is returning some values. Firstly, function bar () is returning 10. Then again function bar () is returning 18. So, as function bar is returning 18 recently to the function foo (), output is recently returned value i.e., 18. As alert method is used to give an output, pop up message is displayed on web page.

**d)**

alert(foo());  
function foo() {  
    var bar = function () {  
        return 13;  
    };  
    return bar();  
    var bar = function () {  
        return 18;  
    };  
}

**Output: 13** with alert pop up message on webpage.

**Reason:** A var bar is a variable function or variable expression, it is returning 13 to bar function and bar function is returning this value 13 to the foo() function which is a parent function. We got an output 13. Hoisting is used here, because function foo() is called before it is defined. So, it is called as hoisting in javascript.