**QUESTION 1**

X: undefined //x at this point is undefined after hoisting before the initialization

a: 8 // a has a value of 8 which is passed a parameter to the function *c().*

b: 8 // b is assigned the value of a which is 8 within the scope of function *f().*

b: 9 // b is later assigned the value of c which is 9 within the scope of function *f().*

b: 10 // finally b has a value of 10 which is got from the global scope

x: 1 // x has a value of 1 which is got from the global scope

**QUESTION 2**

The Global Scope is the scope in which variables or functions are known throughout the application. In other words, variables declared globally have a global scope.

On the other hand, Local Scope is the scope where variables or inner functions are known within the function in which they are declared. Variables or functions declared within a function are scoped to that function.

**QUESTION 3**

1. Do statements in Scope A have access to variables defined in Scope B and C? NO
2. Do statements in Scope B have access to variables defined in Scope A? YES
3. Do statements in Scope B have access to variables defined in Scope C? NO
4. Do statements in Scope C have access to variables defined in Scope A? YES
5. Do statements in Scope C have access to variables defined in Scope B? YES

**QUESTION 4**

81 // myFunction() returns 81 since the value of x is 9 following the global scope.

25 // myFunction() later returns 25 after the value of x has been altered to 5.

**QUESTION 5**

The alert prints out the 10 as the value. This is because, when the function *bar()* is called, in the first phase of compilation, variable *foo* is hoisted and its value is undefined. At execution phase, the *if* condition is true because *foo* is undefined and hence the value of *foo* is set to 10.