**Student Name : Habtom Weldemichael**

**Student ID : 610715**

1. undefined

8

8

9

10

1

1. Global Scope. Any variable that's not inside any function or block (a pair of curly braces), is inside the global scope. ...

Local Scope or Function Scope. Variables declared inside a function is inside the local scope. They can only be accessed from within that function, that means they can’t be accessed from the outside code.

1. The correct answers are (b), (d) and (e)

4. 81

25

1. 10
2. var count= (function(){

var counter=0;

return{

       add:function(){

Return counter+=1;

` },

        reset: function(){

Return counter=0;

}

   }

})();

1. ‘counter’ inside the add function is a free variable. A free variable is a variable which is defined outside that block. Instances created will have a reference to the free variable even after the function is executed (closure).

make\_adder : function(inc){

    return counter+inc;

}

1. Wrap the code in “module pattern” this will avoid all the global variables and functions from exposing to global.

var student=(function(){

var name=”Default”;

var age=0;

var salary=0;

function getAge(){

return this.age;

}

function getSalary(){

return this.salary;

}

function getName(){

return [this.name](http://this.name/);

}

function publicSetAge(newAge){

this.age=newAge;

}

function publicSetSalary(newSalary){

this.salary=newSalary;

}

function publicSetName(newName){

[this.name](http://this.name/)=newName;

}

function publicIncreaseSalary(percentage){

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

}

function publicIncrementAge(){

this.age+=1;

}

return{

setAge: publicSetAge,

setSalary:publicSetSalary,

setName:publicSetName,

increaseSalary:publicIncreaseSalary,

increamentAge:publicIncrementAge

};

})();

student.prototype.address=”default”;

student.prototype.setAddress(newAddress){

this.address=newAddress;

}

student.prototype.getAddress(){

return this.address;

}