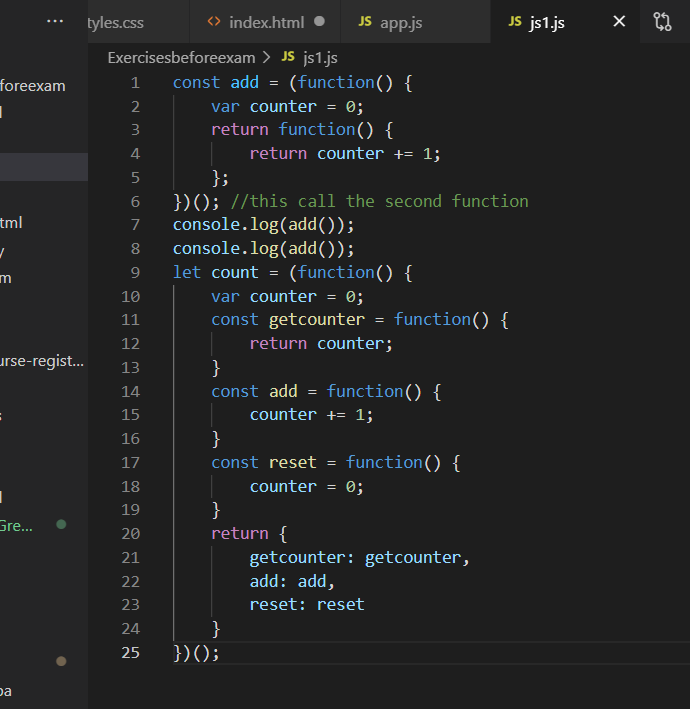
1. undefined
   1. 8
   2. 8
   3. 9
   4. 10
   5. 1
2. Variables that are outside the function are located in the global scope, so this means that we can access it inside the function or outside the function or even inside the block , on the other hand local scope are variables that are declared inside a function or a block, and can only be accessed inside that function or block.

a)No, b)Yes, c)No, d)Yes ,e)Yes

1. a. 81 b. 25
2. 10
3. : //below



1. Counter🡪free variable because is declared and assigned outside the block. Instances created will have access to that free variable by reference even after execution.

make\_adder : function(inc){

return counter+inc;

}

1. We can Wrap the code in **“module pattern”.**

var Employees=(function()

{

var name;

var age;

var salary;

function getAge(){

return this.age;

}

function getSalary(){

return this.salary;

}

function getName(){

return this.name;

}

function publicSetAge(newAge){

this.age=newAge;

}

function publicSetSalary(newSalary){

this.salary=newSalary;

}

function publicSetName(newName){

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

};

})();

1. Employees.prototype.address=”default”

Employees.prototype.setAddress(newAddress){

This.address=newAddress;

}

Employees.prototype.getAddress(){

Return this.address;

}