Lab 7 Assignment

Question 1

Determine what this Javascript code will print out.

Code

```
x = 1;
var a = 5;
var b = 10;
var c = function( a, b, c) {
    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);
```

Output

```
Undefined
8
8
9
10
1
```

Question 2

Define Global scope and Local Scope in javascript.

Any variable declared outside of a function belongs to the global scope, and it is accessible anywhere in your code.

Each function has its own scope, and any variable declared within that function is only accessible from that function and any nested functions.**This is known as local scope** .

Question 3

A: No
B: Yes
C: No
D: Yes
E:Yes
Reason: Variable is accessible in its scope and its nested scopes. But not accessible in its parent scopes and global scope.

Question 4

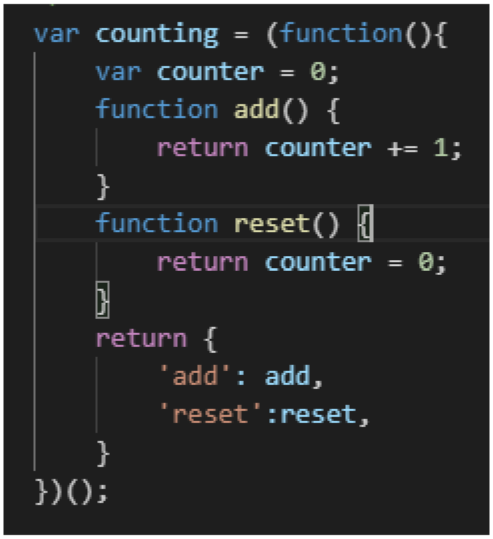
The Output will be:
81
25

Question 5

```
Var foo=1;

function  bar(){

        if(!foo){

                var foo = 10;

        }

        alert(foo);

}
```

```
Result

10 will show in pop up
dialog.
```

Question 6



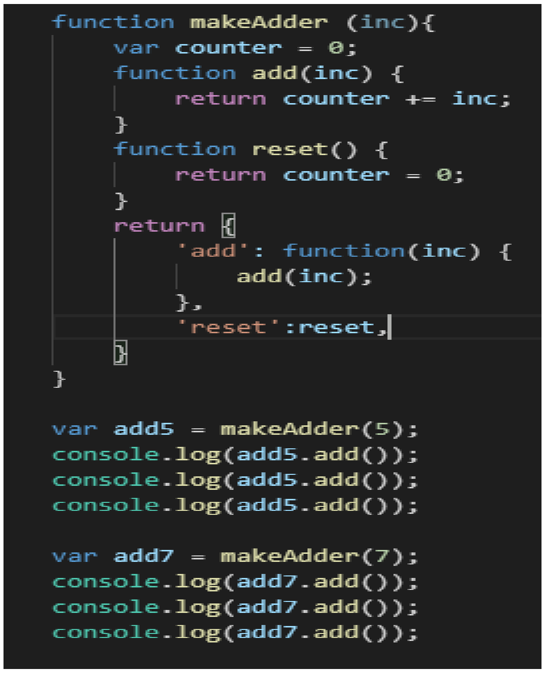
```
var counting = (function(){
    var counter = 0;
    function add() {
        return counter += 1;
    }
    function reset() {
        return counter = 0;
    }
    return {
        'add': add,
        'reset':reset,
    }
})();
```

Question 7

-The free variable is **counter.**

-Free variables are variables that are declared outside a function scope (neither local nor parameters)and are used within the function scope.

Question 8

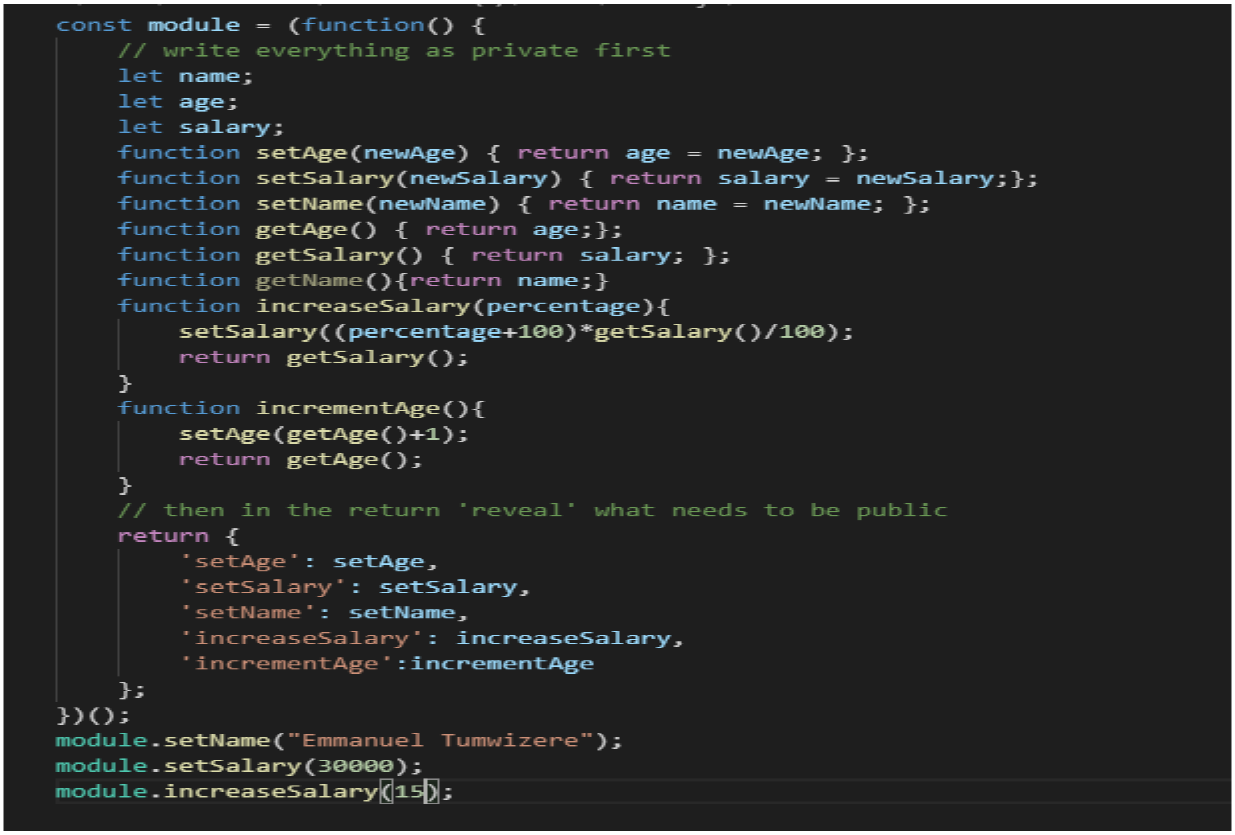


```
function makeAdder (inc){
    var counter = 0;
    function add(inc) {
        return counter += inc;
    }
    function reset() {
        return counter = 0;
    }
    return {
        'add': function(inc) {
            add(inc);
        },
        'reset':reset,
    }
}

var add5 = makeAdder(5);
console.log(add5.add());
console.log(add5.add());
console.log(add5.add());

var add7 = makeAdder(7);
console.log(add7.add());
console.log(add7.add());
console.log(add7.add());
```

Question 9

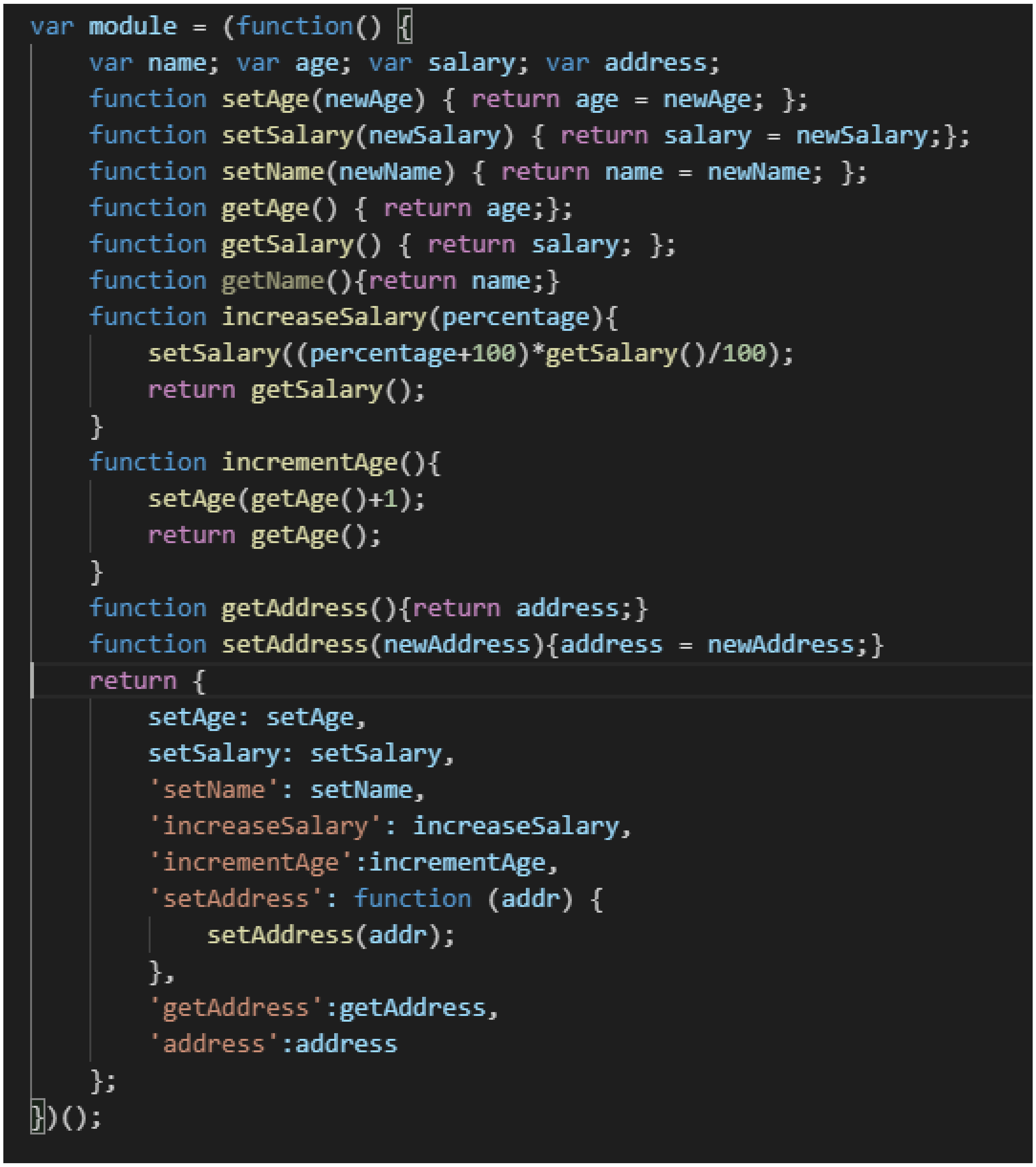
By using IIFE pattern you can wrap all javascript code as a module keep global namespace clean.

Question 10



```
const module = (function() {
    // write everything as private first
    let name;
    let age;
    let salary;
    function setAge(newAge) { return age = newAge; };
    function setSalary(newSalary) { return salary = newSalary;};
    function setName(newName) { return name = newName; };
    function getAge() { return age;};
    function getSalary() { return salary; };
    function getName(){return name;}
    function increaseSalary(percentage){
        setSalary((percentage+100)*getSalary()/100);
        return getSalary();
    }
    function incrementAge(){
        setAge(getAge()+1);
        return getAge();
    }
    // then in the return 'reveal' what needs to be public
    return {
        'setAge': setAge,
        'setSalary': setSalary,
        'setName': setName,
        'increaseSalary': increaseSalary,
        'incrementAge':incrementAge
    };
})();
module.setName("Emmanuel Tumwizere");
module.setSalary(30000);
module.increaseSalary(15);
```

Question 11



```
var module = (function() {
    var name; var age; var salary; var address;
    function setAge(newAge) { return age = newAge; };
    function setSalary(newSalary) { return salary = newSalary;};
    function setName(newName) { return name = newName; };
    function getAge() { return age;};
    function getSalary() { return salary; };
    function getName(){return name;}
    function increaseSalary(percentage){
        setSalary((percentage+100)*getSalary()/100);
        return getSalary();
    }
    function incrementAge(){
        setAge(getAge()+1);
        return getAge();
    }
    function getAddress(){return address;}
    function setAddress(newAddress){address = newAddress;}
    return {
        setAge: setAge,
        setSalary: setSalary,
        'setName': setName,
        'increaseSalary': increaseSalary,
        'incrementAge':incrementAge,
        'setAddress': function (addr) {
            setAddress(addr);
        },
        'getAddress':getAddress,
        'address':address
    };
})();
```