No:6

```
var count = {
    counter: 0,
    add:function () {
        return this.counter += 1;
    },
    reset:function(){
        return this.counter = 0;
    }
};


document.write(count.counter); //0
count.add();
document.write(count.counter);
count.add();
document.write(count.counter);
count.reset();
document.write(count.counter);
```

No:7

Free variables are variables that are neither locally declared nor passed as parameter and they end up within the closure

c1 is free variable

```
var count = {
    counter: 0,
    add:function () {
        c1 =this;
        return c1.counter += 1;
    },
    reset:function(){
        return this.counter = 0;
    }
};

document.write(count.counter);
count.add();
document.write(count.counter);
count.reset();
document.write(count.counter);
```

No:8

```
var count = {
    counter: 0,
    add:function () {
        c1 =this;
        return c1.counter += 1;
    },
    reset:function(){
        return this.counter += 0;
    },

    make_adder:function(n){
        return this.counter+=n;
    },
};

count.make_adder(5);
document.write(count.counter);
count.make_adder(5);
document.write(count.counter);
count.make_adder(5);
document.write(count.counter);
```

No:9

Use IIFE Module pattern, it is to create a new scope and closure around it to keep global name space cleen

```javascript
var Employee = (function() {
    var name='Kadir';
    var age=30;
    var salary=10000;
    function getName(name){
        name=name;
    }
    function getAge(addAge){
        age+=addAge;
    }
    function getSalary(percent){
        salary+=(salary*percent);
    }

    function setName(newName){
        name=newName;
    }
    function setAge(newAge){
        age=newAge;
    }
    function setSalary(newSalary){
        salary=newSalary;
    }
    return {
        increaseSalary: function(p) {
            getSalary(p);
        },
        incrementAge: function(){
            getAge(1);
        },

        name: function(){
            return name;
        },

        age: function(){
            return age;
        },
        salary: function(){
            return salary;
        }

    }

})();

console.log(Employee.name());

Employee.increaseSalary(10)
console.log(Employee.salary());

Employee.incrementAge();
console.log(Employee.age());
```

```
document.write("Name: " + Employee.getName() + ", Age: " + Employee.getAge() + ",
Salary: " + Employee.getSalary());

Employee.setName("Joe");
Employee.setAge(25);
Employee.setSalary(100000);

document.write("<br />");
document.write("Name: " + Employee.getName() + ", Age: " + Employee.getAge() + ",
Salary: " + Employee.getSalary());

Employee.increaseSalary(10);
Employee.incrementAge();

document.write("<br />");
document.write("Name: " + Employee.getName() + ", Age: " + Employee.getAge() + ",
Salary: " + Employee.getSalary());

//test address:

emp1.setAddress("This is his home address");
document.write("Name: " + Employee.getName() + ", Age: " + Employee.getAge() + ",
Salary: " + Employee.getSalary() + ", Address: " + emp1.getAddress());
```