W2D3 Homework

1. Review the `add()` function show below. notice that a module is used to create a private variable called *`counter`*.

   Modify this code to make a revealing module that returns an object with two methods. The returned object should have a `add()` method and a `reset()` method. Where the add method adds 1 to the counter, and the reset method set it back to zero.

   Lastly change the global variable name that will hold this object from `add` to `count`.

```
var add = (function() {
    var counter = 0;
    return function() {
        return counter += 1;
    }
})();
```

2. In the definition of the `add()` function in the code for the question 1, identify the "free" variable. Also write down a definition of what "free" variables are.
3. The `add()` function in question 1 always adds 1 to the *`counter`* each time it is called. Write a `make_adder(inc)` function whose return value is an `add` function with increment value of *`inc`* instead of 1. In other words your function should be able to do the following:

```
var add5 = make_adder(5); // returns a function with its own private counter
add5(); add5(); console.log(add5()); // output is 15

var add7 = make_adder(7); // returns a function with its own private counter
add7(); add7(); console.log(add7()); // output is 21
```

4. Suppose you are given a file of JavaScript code containing a list of many function and variable declarations. All of these function and variable names will be added to the Global JavaScript namespace. What simple modification to the JavaScript file can remove all the names from the Global namespace?
5. Using the Revealing Module Pattern, write a JavaScript definition of a Module that creates an Employee Object with the following fields and methods:

   Private field: name
   Private field: age
   Private field: salary

   Private method: getName()
   Private method: getAge()

Private method: getSalary()

Public method: setName()
Public method: setAge()
Public method: setSalary()
Public method: increaseSalary(percentage) // uses private getSalary()
Public method: incrementAge() // uses private getAge()

6. Write a few JavaScript instruction to make a Module extension to the module made in the previous question. Very important – you are not allowed to modify any of the code to facilitate the extension. Your extension should add a public address field and public methods setAddress(address) and getAddress()