

Final Exam Review

The final exam will consist of those questions:

1. Multiple Choices – 10 questions – total 30 points
(For the multiple-choice questions any number of answers can be correct, possibly none, possibly all, possibly some.)
2. Short answer questions – JSP Elements - 4 questions – total 8 points
3. Programming questions – ? questions
 - a. JSP, Servlet, JSTL – 8 points
 - b. JSP, Servlet, jQuery, AJAX, Module Pattern – 17 points
 - c. Revealing Module Pattern, private/public methods, object literal – 4 points
 - d. JSP, Servlet, jQuery, AJAX, Module Pattern, GSON, JSON – 20 points
4. SCI question – 1 question – 3 points

Location: L113, L115. Total 90 points

Time: May 25, 10am ~ 12:30am, 2.5 hours (Arrive 10 minutes earlier)

Here is a review of Programming Question c:

Module pattern - IIFE

| | | |
|--|---|--|
| <pre>(function(<u>params</u>) { statements; })(<u>params</u>);</pre> |  | <pre>(function(<u>params</u>) { statements; })(<u>params</u>);</pre> |
|--|---|--|

- ▶ **declares and immediately calls an anonymous function**
 - ▶ parens around function are a special syntax that means this is a function expression that will be immediately invoked
 - ▶ – “immediately invoked function expression (IIFE)”
 - ▶ used to create a new scope and closure around it
 - ▶ can help to avoid declaring global variables/functions
 - ▶ used by JavaScript libraries to keep global namespace clean

Module Pattern Example



```
// old: 3 globals
var count = 0;
function incr(n) {
  count += n;
}
function reset() {
  count = 0;
}

incr(4);
incr(2);
console.log("count: " +
  count);

// new: 0 globals
(function() {
  var count = 0;
  function incr(n) {
    count += n;
  }
  function reset() {
    count = 0;
  }

  incr(4);
  incr(2);
  console.log("count: " +
    count);
})();
```

Revealing Module Pattern

```
var Module = (function() {
  var privateMethod = function() {
    // private
  };
  var someMethod = function() {
    // public
  };
  var anotherMethod = function() {
    // public
  };
  return {
    someMethod: someMethod,
    anotherMethod: anotherMethod
  };
})();
```

Creating objects via object literal

```
var name = {  
  'fieldName': value,  
  ...  
  'fieldName': value  
};  
var pt = {  
  'x': 4,  
  'y': 3  
};  
alert(pt.x + ", " + pt.y);
```

- ▶ in JavaScript, you can create a new object without creating a class
- ▶ the above is like a Point object; it has fields named x and y
- ▶ the object does not belong to any class; it is the only one of its kind, a singleton
 - ▶ `typeof(pt) === "object"`