

Scripting Languages G22.3033-002 Summer 2008

hw06

Assigned Th 6/26/2008, due Fr 7/4 at 9pm. 50 points.

<http://www.cs.nyu.edu/courses/summer08/G22.3033-002/>

Homework instructions

Homeworks are due on Fridays at 9pm. This deadline will be strictly enforced.

Email your answers to Robert Soulé robert.soule@gmail.com. Please put your solutions to VBA programming problems in a powerpoint presentation. For all other questions (including programming problems in other languages), just send a simple text file, such as what you get when using Emacs, Vi, Notepad, or the "save as text only" feature in Word.

Please make sure that your code works with the compilers and tools installed at CIWW. Specifically, please test:

JAVASCRIPT

Your JavaScript code must run with **both** Mozilla's Firefox browser **and** with Microsoft's Internet Explorer browser.

PHP Your PHP code must run with PHP 5.1.6 on the CIMS web servers, see <http://www.cims.nyu.edu/systems/userservices/webhosting/>.

PERL Your Perl code must run with Perl 5.8.8 for Linux/x86, for example on doowop1 (see <http://www.cims.nyu.edu/systems/resources/computeservers/>).

VBA Your VBA code must run with Microsoft Office 2003 for Windows on the machines in the labs CIWW 502 or CIWW 624.

Reading assignments

Read for lecture on 7/3:

Jesse James Garrett. Ajax: A New Approach to Web Applications. Essay and FAQ published on the website of Adaptive Path, February 18, 2005. Available at <http://www.adaptivepath.com/ideas/essays/archives/000385.php>.

Concept questions

hw06-1 Closures

(2+4+4+4 = 14 points) Consider the following HTML document with JavaScript.

```
<html><head>
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
  <script>
    function curry(f, x) {
      return function(y) {
        return f(x, y);
      }
    }
  </script>
```

```

</head><body>
  <h1><a href="http://en.wikipedia.org/wiki/Currying">Currying</a></h1>
  <script>
    function add(x,y){ return x + y; }
    var inc = curry(add, 1);
    var double = curry(function(x,y){ return x * y; }, 2);
    document.write(inc(5) + "<br/>\n");
    document.write(double(5) + "<br/>\n");
  </script>
</body></html>

```

- Predict what the code should print. Then run it. What does it print?
- A closure is a function plus an environment. For closure `inc`, what is the function, and what is the environment? And in the environment, what is `f` bound to, and what is `x` bound to?
- For closure `double`, what is the function, and what is the environment? And in the environment, what is `f` bound to, and what is `x` bound to?
- Show how to use function `curry` to create a closure for squaring a number. For example, `square(5)` should return 25.

hw06-2 Prototypes

(2+4+4 = 10 points) Consider the following HTML document with JavaScript.

```

<html><head>
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
  <script>
    function C() { }
    C.prototype.f = function() { return 'Cf'; }
    C.prototype.g = function() { return 'Cg'; }
    C.prototype.h = function() { return 'Ch'; }
    function D() { }
    D.prototype = new C();
    D.prototype.constructor = D;
    D.prototype.g = function() { return 'Dg'; }
    function E() { }
    E.prototype = new D();
    E.prototype.constructor = E;
    E.prototype.h = function() { return 'Eh'; }
  </script>
</head><body>
  <script>
    var x = new D();
    document.write(x.f() + ', ' + x.g() + ', ' + x.h() + "\n<br/>");
    var y = new E();
    document.write(y.f() + ', ' + y.g() + ', ' + y.h() + "\n<br/>");
  </script>
</body></html>

```

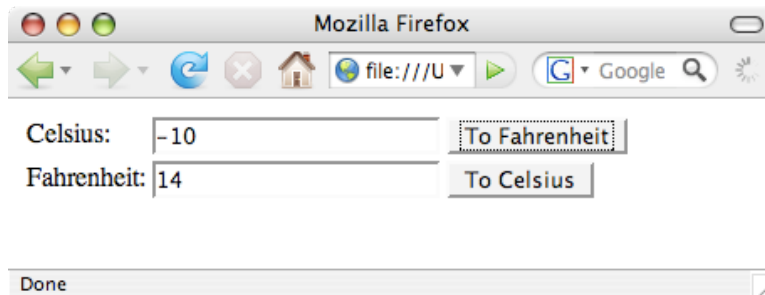
- Predict what the code should print. Then run it. What does it print?

- b. What is the chain of prototypes of object x?
- c. What is the chain of prototypes of object y?

Programming exercises

hw06-3 Temperature conversions (JavaScript)

(12 points) Write an HTML document with a form that has two text input fields and two buttons. When the user enters a value in the “Celsius” field and presses the “To Fahrenheit” button, that should trigger a JavaScript event handler that shows the converted temperature in the “Fahrenheit” input field. Likewise, when the user enters a value in the “Fahrenheit” field and presses the “To Celsius” button, that should trigger a JavaScript event handler that shows the converted temperature in the “Celsius” input field. Below is an example screen shot. Remember to test your code in both the Mozilla Firefox and the Microsoft Internet Explorer web browsers.



hw06-4 Priority queue (JavaScript)

(14 points) Write a constructor `PriorityQueue` that creates an object that supports three methods: `push`, `pop`, and `peek`. A priority queue allows elements to be added in any order, but always returns the smallest element. The `push` method adds an element to the queue. The `pop` method removes and returns the smallest element in the queue. The `peek` method also returns the smallest element in the queue, but does not remove it. If you put your code in a file called `PriorityQueue.js`, then you can test it with the following driver:

```
<html><head>
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
  <script src="PriorityQueue.js"></script>
  <script>
    var q = new PriorityQueue();
    function doPush() {
      var val = document.driver.push.value;
      q.push(val);
      var log = document.getElementById("log");
      log.innerHTML += "push(" + val + ")\n<br/>";
    }
    function doPop() {
      var val = q.pop();
      var log = document.getElementById("log");
```

```

    log.innerHTML += "pop() -&gt; " + val + "\n<br/>";
  }
function doPeek() {
  var val = q.peek();
  var log = document.getElementById("log");
  log.innerHTML += "peek() -&gt; " + val + "\n<br/>";
}
</script>
</head><body>
  <form name="driver">
    <input type="text" name="push">
    <input type="button" value="Push" onclick="doPush();">
    <br/>
    <td><input type="button" value="Pop" onclick="doPop();">
    <br/>
    <td><input type="button" value="Peek" onclick="doPeek();">
  </form>
  <span id="log"></span>
</body></html>

```

The following screen snapshot demonstrates the behavior of a priority queue through a sequence of calls.

