

Scripting Languages G22.3033-002 Summer 2008

hw09

Assigned Th 7/17/2008, due Mo 7/28 at 9pm. 15 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/computerservers/>).

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 7/17:

Andreas Zeller. *From Automated Testing to Automated Debugging*. 2000. Available at <http://www.infosun.fim.uni-passau.de/st/papers/computer2000>

Concept questions

hw09-1 Scientific Method

(3+12 = 15 points) Consider the following buggy Perl script:

```
#!/usr/bin/perl # 1
use strict; use warnings; # 2
# 3
sub compute_avg { # 4
    my @sums; # 5
    my $nrows = $#_; # 6
    for (my $i=0; $i<$nrows; $i++) { # 7
        my $row = $_[$i]; # 8
        for (my $j = 0; $j < @$row; $j++) { # 9
            $sums[$j] += $row->[$j]; #10
```

```

    } #11
  } #12
  my @result; #13
  for (my $i = 0; $i < @sums; $i++) { #14
    $result[$i] = $sums[$i] / $nrows; #15
  } #16
  return @result; #17
} #18
#19
sub to_string { #20
  my $result = ""; #21
  for my $row (@_) { #22
    $result .= " " . $_ for (@$row); #23
    $result .= "\n"; #24
  } #25
  return $result; #26
} #27
#28
our @input = ([2, 1, 3], [0, 1, 1], [4, 1, 2]); #29
print "initial input:\n" . to_string @input; #30
our @expected = ([2, 1, 3], [0, 1, 1], [4, 1, 2], [2, 1, 2]); #31
print "expected output:\n" . to_string @expected; #32
our @avg = compute_avg @input; #33
our @output = @input; #34
push @output, [ @avg ]; #35
print "actual output:\n" . to_string @output; #36

```

- a. Run the script. What does it print?
- b. Debug the script using the scientific method. Your answer should be a debugging log book, similar to the example in the lecture slides. The format should be similar to this:

```

1 Hypothesis (e.g., variable has wrong value in given line)
  Experiment (e.g., run debugger, set breakpoint, inspect value)
  Observation (e.g., the value of the variable at that place)
  Conclusion (e.g., hypothesis verified, falsified, inconclusive)
2 Hypothesis ...
  Experiment ...
  Observation ...
  Conclusion ...
3 ...
...

```

Programming exercises

hw09-2 Delta Debugging

(0 points) When you learn a new language, a good way to make progress is by solving some moderately difficult programming problems. At the same time, when you learn a new algorithm, a good way to understand and remember it is by implementing it. Therefore, I

suggest you combine the two by using one of the new languages you learned in prior lectures to implement one of the debugging algorithms you learned in the most recent lecture.

- Port the `ddmin` algorithm to JavaScript or PHP. You can start from the Perl version on the slides, or you can start from the publicly available Python version by Andreas Zeller.
- Once you have a delta debugging algorithm, the next difficult part left is to automate tests. Create a test harness that provides inputs to a server-side script that would otherwise be typed into a form by the user. Your harness should also check whether the result from the server is correct or faulty.