

Welcome to CS 368!

Introductions, Overview,
Course Mechanics, Resources, etc.

Introductions

Introduction to Perl

Tim Cartwright

Website

<http://pages.cs.wisc.edu/~cs368-3/>

Brief Overview

Course Objectives

- Write basic code in Perl
- Solve real-world problems with scripting languages
- Learn *about* scripting languages
(Perl, Python, Ruby, PHP, JavaScript, ActionScript, ...)

Scripting

- Fast development
- Easy to understand and change
- Abstracts over low-level details
- Pervasive
- Examples:
 - quick (to write) computation
 - data manipulation
 - glue
 - e.g., SpamAssassin, Twitter (Ruby on Rails)

Perl

- Practical Extraction and Report Language
(*Pathologically Eclectic Rubbish Lister?*)
- Introduced 1987
 - Perl 5.8: 2002–2008
 - Perl 5.10: 2007–2009 *version used in class!*
 - ...
 - Perl 5.16: *imminent*
- Widely available
- Great for processing text
- Huge number of independent libraries (CPAN)

Course Philosophy

Learn *a new skill*

Learn *by doing*

Learn *to fish*

Course Mechanics

Homework and Grading

- **Credit**
 - Course offered as credit/no credit
 - All points come from homework (no exam)
- **Homework**
 - Short coding assignment
 - Every day (except last day): 15 total
 - Due by 11:10 a.m. of next class (email tolerated)
 - ***No late assignments accepted at all***
 - Each assignment given 0, 1, or 2 points
 - Need 20 points (67%) to get credit for the course
 - No extra points available

Homework Points

Pts	Reason
2	<ul style="list-style-type: none">• turned in on time, AND• code runs, AND• solution is correct or nearly so, AND• demonstrates real effort
1	<ul style="list-style-type: none">• turned in on time, AND• partial solution, may not actually run, AND• demonstrates some effort (my discretion)
0	<ul style="list-style-type: none">• late, OR• <i>is plagiarized (= Academic Misconduct)</i>, OR• does not demonstrate any real effort

Mailing List

compsci368-3-su12-dhh@lists.wisc.edu

- Goes to @wisc.edu account
- Check spam filters

Office Hours

Computer Sciences 4265 (Tim's office)

Days and times: Doodle poll today!

Other times available by appointment (email)

Course Books

- *Learning Perl* (6th Ed.)
 - *Perl Cookbook* (2nd Ed.)
 - *Programming Perl* (3rd Ed.)
-
- Available FREE online via MadCat
 - Not in the UBS textbook area
 - ***If*** you buy them, consider newest editions

Some Perl Examples

List and Count Directory Entries

```
use File::Basename;
my $entry = $ARGV[0];
print join("\n", glob("$entry/{.,?}*")) . "\n";
print "Number of entries in $entry: ";
print scalar(grep { basename($_) !~ /^\.\.?$/ }
             glob("$entry/{.,?}*"));
print "\n";
```

Average Numbers in Data File Columns

```
my $count = 0;
my @s;
while (<STDIN>) {
    my @w = split;
    $count++;
    for (my $i = 0; $i <= $#w; $i++) {
        $s[$i] += $w[$i];
    }
}

for (my $i = 0; $i <= $#w; $i++) {
    print $s[$i] / $count, "\t";
}
print "\n";
```

Run a System Command As Another User

```
use POSIX "setsid";

my $daemon;
if ($ARGV[0] eq "--detach") {
    shift @ARGV;
    $daemon = 1;
}
my ($user, @cmd) = @ARGV;
die "Can only be run by root\n" if $<;
die "User $user does not exist\n" unless getpwnam($user);

my $new_primary_group = (getpwnam($user))[3];
my $new_secondary_groups = `id -G $user`;
if ($? ne 0) {
    $new_secondary_groups = $new_primary_group;
}

$( = $) = "$new_primary_group $new_secondary_groups";
$< = $> = (getpwnam($user))[2];
if ($daemon) {
    exit if fork;
    setsid;
    STDIN->open("/dev/null");
    STDOUT->open(">>/dev/null");
    STDERR->open(">>/dev/null");
}
exec(@cmd);
exit(1);
```

How to Run Perl

Running Perl

- **Unix/Linux**
 - `perl filename`
 - `chmod 0755 filename`
`./filename`
- **Mac OS X**
 - use Terminal, same as above
- **Windows**
 - download ActiveState Perl
 - not officially supported in the course

Some Basic Syntax

Hello World

```
#!/usr/bin/perl

use strict;
use warnings;

# Everyone's first Perl program
print "Hello, world!\n";
```

Numbers & Math

- literals: **42**, **3.141**, **-6.5e9**, **0377**, **0xff**
- operators: **+** **-** ***** **/** ****** **%** **()**

```
4 + 7           => 11  
17.8 - 3.5     => 14.3  
16 * 0x10      => 256  
2 ** 8         => 256  
10 / 3         => 3.333333...  
10 % 3         => 1  
(2 + 3) * 4    => 20
```


Homework for Day 1

- Go to section website
- Find syllabus
- Find and read homework
- Run the homework script on your own machine and print the output
- Turn in output tomorrow

<http://pages.cs.wisc.edu/~cs368-3/>