Day 10: Correctness

Suggested Reading:

Programming Perl (3rd Ed.) Chapter 20: The Perl Debugger

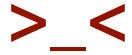
http://perldoc.perl.org/ — Modules Test::Simple, Test::More, Test::Harness

Homework Review

Correctness

I wrote a script!

Did I do it right?



What Does "Do It Right" Mean?

functionality Does it work correctly?

reliability Does it work every time?

usability Is it easy to use?

efficiency Is it fast? Low memory, disk, I/O, ...?

maintainability Is it easy to change?

portability Does it work well everywhere?

(adapted from ISO 9126-1)

What About ... Not So Right?

Failure

Defect

wrong; unexpected behavior

An event: Something went A mistake: Something is likely or certain to cause a failure

Hardware

Cracked solder joint

Network

Flaky router

Data

Data-entry errors

User

Hangover

Software

Bugs!

How to Test / Debug

- "Try it out"
- print() statements / logging
 - see, e.g., Log::Log4perl
- Debugger
- Automated testing
- Code review
- Formal analysis

Manual Testing

Manual Debugging: print()

```
my \$DEBUG = 1;
sub convert {
    my ($from, $to, $value) = @ ;
    print "from = $from\n" if $DEBUG;
    print "to = $to\n" if $DEBUG;
    my $meters = $value * $UNITS{$from};
    print "meters = $meters\n" if $DEBUG;
    my $result = $meters / $UNITS{$to};
    print "result = $result\n" if $DEBUG;
    return $result;
```

Debuggers

Common debugger features:

- View code
- Step through (running) code, line-by-line
- Examine variables
- Run and stop at breakpoints
- Stack traces
- Watch points

The Perl Debugger

```
List/search source lines:
                                      Control script execution:
 l [ln|sub] List source code
                                                    Stack trace
             List previous/current line s [expr]
  - or .
                                                    Single step [in expr]
           View around line
 v [line]
                                        n [expr]
                                                    Next, steps over subs
 f filename View source in file
                                        <CR/Enter> Repeat last n or s
                                                    Return from subroutine
 /pattern/ ?patt? Search forw/backw
                                        c [ln|sub] Continue until position
             Show module versions
Debugger controls:
                                                   List break/watch/actions
             Set debugger options
 o [...]
                                        t [expr] Toggle trace [trace expr]
 <[<]|{[{]|>[>] [cmd] Do pre/post-prompt b [ln|event|sub] [cnd] Set breakpoint
  ! [N|pat] Redo a previous command
                                        B ln|*
                                                    Delete a/all breakpoints
                                        a [ln] cmd Do cmd before line
 H [-num] Display last num commands
 = [a val] Define/list an alias
                                        A ln|*
                                                    Delete a/all actions
 h [db cmd] Get help on command
                                        w expr Add a watch expression
             Complete help page
                                        W expr|*
                                                    Delete a/all watch exprs
  h h
  |[|]db cmd Send output to pager
                                        ![!] syscmd Run cmd in a subprocess
 a or ^D
             Ouit
                                        R
                                                    Attempt a restart
Data Examination:
                             Execute perl code, also see: s,n,t expr
                     expr
                Evals expr in list context, dumps the result or lists methods.
 x|m expr
                Print expression (uses script's current package).
 p expr
                List subroutine names [not] matching pattern
 S [[!]pat]
 V [Pk [Vars]] List Variables in Package. Vars can be ~pattern or !pattern.
 X [Vars]
                Same as "V current package [Vars]". i class inheritance tree.
                List lexicals in higher scope <n>. Vars same as V.
 y [n [Vars]]
       Display thread id
                            E Display all thread ids.
For more help, type h cmd letter, or run man perldebug for all docs.
```

Automated Testing

Automated Testing

- Write software to test (other) software
- Humans vs. machines
- Types of automated tests
 - Unit tests
 - Functional tests
 - Performance tests

Test::Simple

Run a . t file to test a module

```
#!/usr/bin/perl
use strict; use warnings;
use SomeModule; # to be tested
use Test::Simple tests => 2;
ok(is doing ok(), 'doing ok');
ok(the result() == 7, 'value ok');
```

Test::More

```
use Test::More tests => 6;
ok(is ok(), 'is OK');
is($the answer, 42, 'answer ok');
isnt(exit status(), 1, 'syscall');
like($name, qr/tim/i, 'good name');
unlike($result, qr/error/i, 'test');
diag("current value of name = $name");
SKIP: {
  skip 'not available', 1 if ...;
  ok(...);
};
```

Testing a Standalone Script

```
use Getopt::Long;
GetOptions('test' => \&run tests);
# main script & subroutines here
sub run tests {
  require Test::More;
  Test::More->import;
  plan(tests => nnn);
  # test subroutines here
  exit 0;
```

Unit Testing Tips

- Test logical chunks of code usually subroutines
- Aim for reasonable coverage
- Run often!
 - After every (significant) change
 - Before you use, hand in, commit, ...
- Encode past failures as tests

Test-First Development

Radical idea: Write tests FIRST

- Then write code until tests pass
- Clarifies and documents design
- And of course... is useful for testing!

http://junit.sourceforge.net/doc/testinfected/testing.htm

Wrap Up

Other Scripting Languages

- "Try it out" and printing/logging always work
- Most have debuggers and/or interactive modes
- Unit testing:
 - Most others are based on jUnit
 - Expect similar and richer assertions
 - Introspection rocks!

Homework

- Write a couple of simple but non-trivial functions
- Write unit tests against them
- Make script work in "test" and normal modes