1ST Class – Wednesday evening, January 7, 2015

Instructor: Chris Oson – [coson@coleman.edu](mailto:coson@coleman.edu)

Perl – A very interesting language. It is very permissive. It can be a very eloquent language.

Can be a compact, terse language.

Perl – a backronym. Practical Extraction and Report Language. 1987. 1988… v5 Overhaul w/ OOP 1991

1996-1997 hay day of CGI – Common Gateway Interface, uses Perl for dynamic content.

1998-1999 PHP, Python, ASP, ColdFusion (They were better alternatives to CGI).

Unix had built-in tools sed, ack, grep.

Unix shells (desktop): bash, bourne, aix – korn, csh, tsch.

We’re going to cover the Unix tools: sed, ack, grep

AciveState.com

CPAN

Perl 5.1.16 (Latest)

Installed with the Interpreter and the Perl Package Manager (PPM) in C:\Perl with perl.exe

IDE: Notepad++

If not using an IDE we will run our programs at a command line.

Store your Perl scripts at C:\coleman\com330> with files Assignment1.pl

Method A: type >c:\Perl\perl.exe Assignment1.pl

Method B: setup the Environment variable %PATH% to append the directory path.

Method C: use an IDE: Komodo Dragon, Padre (advantage)

Programming Assignment 1 (15 points) COM330-08427-01.pl

Programming Assignment 2 (20 points) COM330-08427-02.pl

Programming Assignment 3 (15 points) COM330-08427-03.pl

Midterm (25 points)

Final (25 points)

100 Points, No quizzes, no reviews.

Submitting assignments: Filename format: *COM330-xxxxxx-yy.pl, zip*, 7z, rar, tar (make sure it compiles)

WEBCLASS:

1. Which version of Perl? 5.1.16

2. Tools to write Perl scripts On Windows or Unix, use Eclipse or Visual Studio.

Perl has scalar values (single value) like numbers, symbols, characters, string. As opposed to an aggregate.

It uses a sigil: ‘$’ (for scalar values)

Treats variables as dynamic data types.

Semicolon is necessary for statement end of line.

Print function:

Print $val;

#!usr/bin/perl # This comment line is ignored in Windows

# but in Unix it specifies the location of the interpreter.

# And it is the only statement line that does not have a semicolon.

For strings you can use single quotes ‘or double quotes “

- but using single quotes ‘the \n newline is not recognized.

- and using double quotes ‘the \n newline is recognized as a new line.

Variable interpretation e.g. $new = “This is \$$val\n”;

Special environment variable:

$\=”\n”;

Backlash escapes: \n, \r, \t, \f, \b, \\, \l, \L, \u, \U

Get user input: Make a prompt:

<STDIN>, <STDOUT>, <STDERR>

#!usr/bin/perl

$\ = “\n”;

$prompt = “Please enter something: “;

Print $prompt;

$input = <STDIN>;

# $input = “hello\n”

chomp($input); # CHOMP removes one character at the end of the line.

# $input = “hello”

**TIMTOWTDI** – there is more than one way to do it. But the Perl way is to do this:

chomp($input = <STDIN>);

**Strings:**

If ($sFirst eq $sSecond) {

}

…

**Numbers:**

If ($First == $Second) {

}

… # The If statements require curly braces all the time (even for a single statement).

String concatenation with the period ‘.’ sign: print $word1 . $word2;

Boolean values:

$name = ‘joe’;

$is\_bigger = $name gt ‘fred’ # Perl does not have a dedicated True/False

If value is a number, a 0 is False, and everything else is true.

If value is a string, an empty string is False, and everything else is true.

Assignment 1: A lot of work!

#!usr/bin/perl

$\ = “\n”;

system(“cls”); # to clear the screen in windows

Perl does not have a switch statement but it does have an elsif (lack of the e)

If ($input < 0) {

Print “1 – The number is less than 0”

} elsif ($input > 0 && $output <= 25) {

Print “1 – The number is in our range”;

} else {

Print “1 – The number is greater than 25”;

}

The opposite of the if statement with ‘unless’

$test = 25 < 15;

unless ($test) {

Print “Execute the false condition only”;

}

print “2 - The number is less than 25” if ($input <= 25);

**While Loop:**

#!usr/bin/perl

$\ = “\n”;

system(“cls”);

($average, $total, $count) = (0, 0, 0);

$grade = ‘’;

while ($grade ne ‘q’) {

print “Enter a grade or ‘q’ to quit: “;

chomp($grade = <STDIN>);

if ($grade ne ‘q’) {

$count += 1;

$total += $grade;

}

}

$count ||= 1; # Avoid division by zero if nothing entered

$average = $total / $count;

Print “The average is $average”;

||= Or equals – done a lot in JQuery.

**For Loop:**

#!usr/bin/perl

$\ = “\n”;

system(“cls”);

$greyRow = “#CCCCCC”;

$whiteRow = “#FFFFF”;

$line = “<tr style=’%s’><td>Row</td><td></td></tr>”;

printf(“%d + %d = %d\n”, 5, 5, 5 + 5)” # format specifiers

$formatted\_printf = sprint(“%d + %d = %d\n”, 7, 7, 7 + 7)”

# like printf but it saves the string and assigns it to another variable.

Print $formatted\_printf;

$table = “<table>”;

print “<table>\n”

for ($i = 0, $i < 10; $i++) {

if ($i % 2) == 0) {

$table .= “<tr style=\”” . $grewRow . “\”><td>Row</td><td>” . ($i + 1) . “</td></tr>\n”;

} else {

$table .= “<tr style=\”” . $whiteRow . “\”><td>Row</td><td>” . ($i + 1) . “</td></tr>\n”;

}

$table .= ((i % 2) == 0)

? sprintf($line, $greyRow, ($i + 1))

: sprintf($line, $whiteRow, ($i + 1));

}

$table .= “</table>\n”;

Replaced by template Engines to merge data with perl script. Then there were Smarty templates. With a tool called Mason.

We can clean this up (like in PHP) by using the printf function.

What drives your decisions of what eloquence you choose…

comes down to knowing if ‘you’ will be maintaining it.

print “</table>”

**String Functions**

Next week – we’ll talk about Transliteration

#!usr/bin/perl

$\ = “\n”;

system(“cls”);

print “\033[2J; # Clears the screen

print “\033[0;0H”; # Jump to 0,0

$input = “This is a test”;

$pattern = ‘is’;

$notFound = ‘xxx’;

print “The number of characters in \$input is: “ . length $input;

print “The string in lower case is: “ . lc($input);

print “The string in upper case is: “ . uc($input);

# Is a string pattern inside another string?

print “The pattern ‘$pattern’ is in ‘$input’ at position: “ . index($input, $pattern);

print “The pattern ‘$notFound’ is in ‘$input’ at position: “ . index($input, $notFound);

print “The pattern ‘$pattern’ is in ‘$input’ at position: “ . rindex($input, $pattern);

**Substring Functions**

$save1 = “The substring “ . substr($input, 2); # ‘is’ in this is at position 2

$save2 = “The substring “ . substr($input, 4, 5); # Returns the next 5 characters.

Print $save1;

Print $save2;

Two overloads.

Gives the position it was found. If it is not found it returns -1.

Perldoc.perldoc.org has good documentation.

2ND Class – Wednesday evening, January 14, 2015

Instructor: Chris Oson – [coson@coleman.edu](mailto:coson@coleman.edu)

Perl has 3 variable types:

scalar variables – Last week. The sigel is $

arrays – tonight. Different in Perl tan other languages

hash – next week.

**Arrays:** The sigel is @:

@simple = (7, 8, 9);

$simple[0] = 7;

$simple[1] = 8;

$simple[2] = 9;

Print $simple[0];

Arrays vs Lists:

#!/user/bin/perl

@array = (1, 2, 3);

print $separator;

@emptyArray = ();

push @emptyArray, $\_ for (1..15);

foreach (@emptyArray) -1] = 80{

print;

}

# another example

push @array, $\_ for 4..8;

printf "value %d\n", shift @array while @array;

# shift - grab first item and remove it. It changed the contents of the array.

@array = ('Larry', 'Curly', 'Moe');

# Equivalent statement...

$size = scalar(@array);

while ($size > 0) {

printf "value %d\n";

shift @array;

$size = scalar(@array);

print "My stooges are: " . join(" ", @array) . "\n";

# join( delimiter, the array ) it converts Ints into Strings so do last, after mathematical operations

# shift removes the item.

# unshift puts an item (back) in.

unshift(@array, 28);

unshift(@array, 56);

unshift(@array, 28);

# Arrays are permitted to have duplicate items

# Arrays use parentheses to create elements but list use brackets

# Arrays use brackets to access the elements

$ref = [ 11, 22, 33 ];

push @$ref, 44, 55, 66;

print "$\_\n" for @$ref;

print "$\_\n" for (25..30);

$op = 1 == 1;

$types = ("abc", 3.33, $op, 555);

print \*$\_\n" for @types;

# abc

# 3.33

1 # when true

# when false

# 555

@words = ("Oh", "say", "can", "you");

$refToArray = \@words;

# I'm not escaping the @ symbol. I'm referencing the array.

# qw returns an array of tokens

@moreWords = qw(Today is Wednesday January 14, 2015);

print "$\_\n" for @moreWord;

# the $# returns the count of characters

print $#moreWords;

for $1 (0..$#moreWords) {

}

print $moreWords[$#morewords + 1];

$moreWords[100] = "Hello';

print $\_\n" for @moreWords;

print "There are " . scalar(@moreWords) . " words in the array.\n";

# use qw with parentheses and with multip[le operators e.g. / forward slash

@moreWords = qw'"San Diego" Today is Wdednesday January 14, 2015 $data');

@moreWords = \qw'"San Diego" Today is Wdednesday January 14, 2015 $data');

print "Enter some input: ";

# input from IO with <>

while (<>) {

for $word (split) {

print $word "\n";

}

}

# input from IO with <>

while (<>) {

for (split) {

print;

}

}

# input from IO with <>

while (<>) {

for (split(',' $\_)) {

print;

}

}

# JavaScript is huge on the scope also.

# Dumper module - Used as a debugging tool

use Data::Dumper qw(Dumper);

$\ = "\n"

system("cls");

print Dumper \@words;

# Format of Input File:FirstName, LastName, StudentID, Mod #, Start Date

$inputLine = "Joe:Smith:50432:1:03-16-2014";

($firstname, $lastName, $studentID, $modNo, $startDate) = split /:/, $inputLine; # multiple variables

$outputLine = sprintf("%s %s - %s", $firstName, $lastName, $studentID, $modNo, $startDate);

# split with string types first.

# %5.1f

@names = ("One", "Two", "Three");

print $names;

print "Using undef: ";

print "We have something " if defined($firstName);

print "There is nothing " if unless(defined($gradDate));

$sample = (5, 6, 7, 8, 9, 10, 11, 12, 13);

print $sample[0];

print $sample[-1];

# splice array, offset, length

@newSample = splice @sample, 1, 5;

print "\@newSample: " . scalar(@newSample);

print $separator;

print for @newSample;

print $separator;

# Perl subroutine - Adds two numbers

sub AddTwoNumbers

{

($first, $second) = @\_;

return $first + $second

}

# Perl subroutine - Square a number

sub AddTwoNumbers

{

$number = shift;

return $number \* $number;

}

@moreNumbers = (10..20)

for ($i = 0; $i <= $#moreNumbers; $i++) {

print "$i - " . SquareNumber($moreNumbers[$i]);

}

# Perl's map function -

@squareNumbers = map { $\_ \* $\_} @moreNumbers;

for ($i = 0; $i <= $#squareNumbers; $i++) {

print "$i - " . $squareNumber[$i]);

}

# String manipulation with the map function to create a lookup table

@randomNumbers = (44, 903, 231, 5722, 474, 90, 471);

@oddNumbers = map { ($\_ & 2) == 2} @randomNumbers;

print for @oddNumbers;

1 11 11 1 # Returns the boolean value - Gotcha!!!

# grep utility is used to search text, recursively. -

# - Anonymous functions, Closest thing to Lambda expressions

@numbers = qw(8 2 5 3 4 34 22 11 17 9);

@numbersLargerThanTEN = grep {$\_ > 10} @numbers;

print for @numbersLargerThanTEN;

$# - largest index in the list

# perl -le 'print for \( 4, 5, 6 );'

# perl -le "print for \( 4, 5, 6 );"

# ^(.\*)

@results = split/' ', $val/;

# Rule #1

# 1. An array is a variable but a list is not.

# 2. Arrays content can be changed but a list's contents cannot.

# 3. Arrays can have names but a list cannot.

# 4. Arrays can have sigels but a list may not.

# 5. Arrays can be referenced but a list cannot be.

Next Week: Dictionaries, Associated Arrays, Classes.

3RD Class – Wednesday evening, January 21, 2015

Instructor: Chris Oson – [coson@coleman.edu](mailto:coson@coleman.edu)

**Topic:**

Data structure: Perl Hash (Formerly known as an ‘associated array.’

In Perl we declare an empty hash:

%areaCodes = (“619”, “San Diego”,

“415”, “San Francisco”,

“707”, “Santa Rosa”);

Print $areaCodes{“415”};

I can overwrite the value like this:

$areaCodes{“619”} = “Downtown San Diego”;

Note: We’ll be using the hash in assignment 2.

Use the fat-comma operator, like this:

%areaCodes = {"619" => "San Diego",

"714" => "Orang County",

"415" => "San Francisco",

"707" => "San Rosa");

Using a new directive called the switch module:

Perl is a loosely typed language so I will use the strict

Requires an explicit

use strict # and preface my variables by adding ‘my’

Use for the parameters passed to a sub-routine.

Passing Arrays into sub-routines:

You are only passing a reference to the first value in the array.

With a hash you won’t know what order you will receive the data back.

While Loop – access with each

For Loop – access by the keys

Regular Expressions

Perl has a transliteration operator

I.e., we have some lower-case data.

Three ways to learn if a pattern is inside another pattern [in strings]:

Find, Substring, Index. I.e.,

$input = “This is a test.”

print index $input, “is”; # “2”

print substr $input, 5; # “is a test.”

print substr $input, 5, 7; # “is a te”

=~ another way to assign a value to a variable.

$expr = "This is a test"

print $expr =~ tr/iat/943/; # inline transliteration

$expr =~ tr/iat/943/;

print $expr;

tr works against the $\_ default variable

$\_ = "This is a test"

tr//\_/g;

Substitution

s/old/new/ig;

Assignment 2 – String manipulation

4TH Class – Wednesday evening, January 28, 2015

Instructor: Chris Oson – [coson@coleman.edu](mailto:coson@coleman.edu)

**Topic:**

Data structure: Perl Hash (Formerly known as an ‘associated array.’

Returning to last week’s problem with passing parameters:

Translational operator and the substitution operator work on the default variable:

Regular Expressions:

Will require the ‘use strict’

System variables:

$\

$\_

@\_

New tonight:

$^O – identify the operating system

system(“clear”);

system $^O

Print tr/A-Z/a-z/;

Print lc;

$\_ = “this is a test”;

print s/is/xx/; # thxx is a test

print s/is/xx/g; # thxx xx a test

$\_ = “THIS IS a test”;

print s/is/xx/ig; # THxx xx a test

# If you don’t want to use the default variable…

my $sentence = “I would like to go to london this year”;

print “\$sentence before: $sentence”;

$sentence =~ s/London/London/;

#IMPORTANT

# I’m using the ‘~’ to change the contents of an existing variable not the default variable.

print “\$sentence after: $sentence”;

alaska

alabama

# We need a file handle and n input file

@emptyList = ();

open FH, “inputfile.txt” or die(“Can’t open: $! $.”);

while (<FH>) {

# don’t need the: $temp = $\_;

$propercase = ucfirst;

push(@emptyList = $propercase);

}

# $! Tells what the problem was

# $. Tells where the problem was by the line number

# the die function to terminate the execution of a script

Homogeneous – An array in another programming language (Java, C#)

Heterogeneous – An array of different types (Perl)

# Grep searches for patterns

my @h\_array = (1, “Test”, 55, “foo”, 65, 20);

my @has\_digits = grep /-?\d+(\.\d+)/, #h\_array; # (1, 655, 65, 20)

\ – the class character

\d – any digit 0-9

+ – meta-character that has special meanings. 1 or more occurrences of the previous character

? – the previous character is optional

. – any single-character

() – to group or remembering data

$ – end of the line or end of the current value

Example:

bled

bleed

^ble+d

5TH Class – Wednesday evening, February 4, 2015

6TH Class – Wednesday evening, February 11, 2015

See Exercise2 for in-class instructions.

Web HTTP Request/Response Model

Request becomes a ‘Virtual Address’ and a request header as a text message.

Web server translates it to a physical address.

Then looks for the resource (web page)

Then returns the response back to the user.

The response has two parts:

The header and the payload:

- Payload is typically an HTML document.

- Response Header contains: metadata

Mime type.

Content length

Cookie data

Status code = 200 OK

And more data…

Templating a System

HTML opened in DreamWeaver opened by designers

Code Files opened by programmers

Tmpl Templating files opened by developers

Smarty templates in 2012.

With .NET ASP. Code behind model.

JavaScript

Angular

2007-2008. Smalltalk ‘MVC’

Laravel, PHP, Zend.

-----

Tracking who comes to your website.

There is Google analytics

print phpinfo();

7TH Class – Wednesday evening, February 19, 2015

I was told to follow the tutorial instructions at <http://www.cgi101.com/> I did the five steps below, I couldn’t get the server to start, so I undid step 1 to put it back. DirectoryIndex and CGI handler is added.

At <http://www.cgi101.com/learn/connect/winxp.html>, it says, Configuring Apache – you need to modify the Apache configuration file to tell it where your pages are, and enable CGI programs.

I have a Bitnami WAMP stack and it’s a little different:

In the file: C:\Bitnami\wampstack-5.4.32-0\apache2\conf\httpd.conf I found:

# User home directories

#Include conf/extra/httpd-userdir.conf

*1st: I uncommented that include line of code.*

In the file: C:\Bitnami\wampstack-5.4.32-0\apache2\conf\extra\httpd-userdir.conf I found:

#

# UserDir: The name of the directory that is appended onto a user's home

# directory if a ~user request is received. Note that you must also set

# the default access control for these directories, as in the example below.

#

UserDir "My Documents/My Website"

#

# Control access to UserDir directories. The following is an example

# for a site where these directories are restricted to read-only.

#

<Directory "C:/Documents and Settings/\*/My Documents/My Website">

AllowOverride FileInfo AuthConfig Limit Indexes

Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

Require method GET POST OPTIONS

</Directory>

*2nd: I changed UseDir to use my existing directory:*

UserDir "My Documents/My Web Sites"

*3rd: I commented out the directory block and added this code and updated the options line,* per the CGI tutorial at <http://www.cgi101.com/learn/connect/winxp.html>:

# From http://www.cgi101.com/learn/connect/winxp.html

# for COM330 Perl CGI Programming

#<Directory "C:/Documents and Settings/\*/My Documents/My Website">

#

<Directory "C:/Users/Mike/Documents/My Documents/My Web Sites">

AllowOverride FileInfo AuthConfig Limit

Options MultiViews Indexes SymLinksIfOwnerMatch Includes ExecCGI

<Limit GET POST OPTIONS PROPFIND>

Order allow,deny

Allow from all

</Limit>

<LimitExcept GET POST OPTIONS PROPFIND>

Order deny,allow

Deny from all

</LimitExcept>

</Directory>

*4th: In the httpd.conf file, I commented out DirectoryIndex and added this to add the index.cgi addition:*

#DirectoryIndex index.html index.php

DirectoryIndex index.html index.html.var index.cgi

*5th: In the httpd.conf file, I added .cgi .pl and the next line*

# From http://www.cgi101.com/learn/connect/winxp.html

# for COM330 Perl CGI Programming, added .cgi .pl and the next line.

AddHandler cgi-script .cgi .pl

AddHandler server-parsed .html

To install the DBI I went to <https://metacpan.org/pod/Bundle::DBI> and selected the latest which is DBI-1.633 and then run at the command prompt:

perl -MCPAN -e 'install Bundle::DBI'

and results:

t/destroy.t ........... ok

t/downgrade.t ......... ok

t/file\_magic.t ........ ok

t/forgive.t ........... ok

t/freeze.t ............ ok

t/integer.t ........... ok

t/interwork56.t ....... ok

t/just\_plain\_nasty.t .. ok

t/lock.t .............. ok

t/malice.t ............ 1/420 Can't open 'storable-testfile.1092': Permission denied at t/testlib.pl line 25.

Can't unlink 'storable-testfile.1092': No such file or directory at t/testlib.pl line 8.

END failed--call queue aborted at t/malice.t line 25.

# Looks like you planned 420 tests but ran 27.

# Looks like your test exited with 2 just after 27.

t/malice.t ............ Dubious, test returned 2 (wstat 512, 0x200)

Failed 393/420 subtests

t/overload.t .......... ok

t/recurse.t ........... ok

t/restrict.t .......... ok

t/retrieve.t .......... ok

t/robust.t ............ ok

t/sig\_die.t ........... ok

t/store.t ............. ok

t/threads.t ........... ok

t/tied.t .............. ok

t/tied\_hook.t ......... ok

t/tied\_items.t ........ ok

t/utf8.t .............. ok

t/utf8hash.t .......... ok

t/weak.t .............. ok

Test Summary Report

-------------------

t/malice.t (Wstat: 512 Tests: 27 Failed: 0)

Non-zero exit status: 2

Parse errors: Bad plan. You planned 420 tests but ran 27.

Files=34, Tests=2221, 8 wallclock secs ( 0.45 usr + 0.20 sys = 0.66 CPU)

Result: FAIL

Failed 1/34 test programs. 0/2221 subtests failed.

dmake.exe: Error code 255, while making 'test\_dynamic'

AMS/Storable-2.51.tar.gz

C:\Perl64\site\bin\dmake.exe test -- NOT OK

//hint// to see the cpan-testers results for installing this module, try:

reports AMS/Storable-2.51.tar.gz

Running install for module 'Net::Daemon'

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/M/MN/MNOONING/Net-Daemon-0.48.tar.gz

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/M/MN/MNOONING/CHECKSUMS

Checksum for C:\Perl64\cpan\sources\authors\id\M\MN\MNOONING\Net-Daemon-0.48.tar.gz ok

Net-Daemon-0.48/

Net-Daemon-0.48/ChangeLog

Net-Daemon-0.48/lib/

Net-Daemon-0.48/lib/Net/

Net-Daemon-0.48/lib/Net/Daemon/

Net-Daemon-0.48/lib/Net/Daemon/Log.pm

Net-Daemon-0.48/lib/Net/Daemon/Test.pm

Net-Daemon-0.48/lib/Net/Daemon.pm

Net-Daemon-0.48/Makefile.PL

Net-Daemon-0.48/MANIFEST

Net-Daemon-0.48/MANIFEST.SKIP

Net-Daemon-0.48/META.yml

Net-Daemon-0.48/README

Net-Daemon-0.48/regexp-threads

Net-Daemon-0.48/t/

Net-Daemon-0.48/t/base.t

Net-Daemon-0.48/t/config.t

Net-Daemon-0.48/t/fork.t

Net-Daemon-0.48/t/forkm.t

Net-Daemon-0.48/t/ithread.t

Net-Daemon-0.48/t/ithreadm.t

Net-Daemon-0.48/t/loop-child.t

Net-Daemon-0.48/t/loop.t

Net-Daemon-0.48/t/server

Net-Daemon-0.48/t/single.t

Net-Daemon-0.48/t/thread.t

Net-Daemon-0.48/t/threadm.t

Net-Daemon-0.48/t/unix.t

Configuring M/MN/MNOONING/Net-Daemon-0.48.tar.gz with Makefile.PL

Checking if your kit is complete...

Looks good

Generating a dmake-style Makefile

Writing Makefile for Net::Daemon

Writing MYMETA.yml and MYMETA.json

MNOONING/Net-Daemon-0.48.tar.gz

C:\Perl64\bin\perl.exe Makefile.PL INSTALLDIRS=site -- OK

Running make for M/MN/MNOONING/Net-Daemon-0.48.tar.gz

cp lib/Net/Daemon/Log.pm blib\lib\Net\Daemon\Log.pm

cp lib/Net/Daemon/Test.pm blib\lib\Net\Daemon\Test.pm

cp lib/Net/Daemon.pm blib\lib\Net\Daemon.pm

MNOONING/Net-Daemon-0.48.tar.gz

C:\Perl64\site\bin\dmake.exe -- OK

Running make test

C:\Perl64\bin\perl.exe "-MExtUtils::Command::MM" "-MTest::Harness" "-e" "undef \*Test::Harness::Switches; test\_harness(0, 'blib\lib', 'bl

ib\arch')" t/\*.t

t/base.t ........ ok

t/config.t ...... ok

t/fork.t ........ skipped: (no reason given)

t/forkm.t ....... skipped: (no reason given)

t/ithread.t ..... Perl exited with active threads:

t/ithread.t ..... 1/? 0 running and unjoined

1 finished and unjoined

0 running and detached

t/ithread.t ..... ok

t/ithreadm.t .... Dubious, test returned 255 (wstat 65280, 0xff00)

Failed 9/10 subtests

t/loop-child.t .. ok

t/loop.t ........ ok

t/single.t ...... ok

t/thread.t ...... Using Perl version 5.20

Old threads style supplanted by ithreads after Perl version 5.10

t/thread.t ...... skipped: Using Perl version 5.20

t/threadm.t ..... ok

t/unix.t ........ skipped: (no reason given)

Test Summary Report

-------------------

t/ithreadm.t (Wstat: 65280 Tests: 1 Failed: 0)

Non-zero exit status: 255

Parse errors: Bad plan. You planned 10 tests but ran 1.

Files=12, Tests=39, 83 wallclock secs ( 0.09 usr + 0.00 sys = 0.09 CPU)

Result: FAIL

Failed 1/12 test programs. 0/39 subtests failed.

dmake.exe: Error code 255, while making 'test\_dynamic'

MNOONING/Net-Daemon-0.48.tar.gz

C:\Perl64\site\bin\dmake.exe test -- NOT OK

//hint// to see the cpan-testers results for installing this module, try:

reports MNOONING/Net-Daemon-0.48.tar.gz

Running install for module 'RPC::PlServer'

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/M/MN/MNOONING/PlRPC/PlRPC-0.2020.tar.gz

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/M/MN/MNOONING/PlRPC/CHECKSUMS

Checksum for C:\Perl64\cpan\sources\authors\id\M\MN\MNOONING\PlRPC\PlRPC-0.2020.tar.gz ok

PlRPC/

PlRPC/lib/

PlRPC/lib/RPC/

PlRPC/lib/RPC/PlClient/

PlRPC/lib/RPC/PlClient/Comm.pm

PlRPC/lib/RPC/PlServer/

PlRPC/lib/RPC/PlServer/Comm.pm

PlRPC/lib/RPC/PlServer/Test.pm

PlRPC/lib/RPC/PlClient.pm

PlRPC/lib/RPC/PlServer.pm

PlRPC/lib/Bundle/

PlRPC/lib/Bundle/PlRPC.pm

PlRPC/ChangeLog

PlRPC/Makefile.PL

PlRPC/MANIFEST

PlRPC/META.yml

PlRPC/README

PlRPC/t/

PlRPC/t/base.t

PlRPC/t/client.t

PlRPC/t/compress.t

PlRPC/t/crypt.t

PlRPC/t/lib.pl

PlRPC/t/methods.t

PlRPC/t/server

Configuring M/MN/MNOONING/PlRPC/PlRPC-0.2020.tar.gz with Makefile.PL

Checking if your kit is complete...

Looks good

Warning: prerequisite Net::Daemon 0.13 not found.

Generating a dmake-style Makefile

Writing Makefile for RPC::PlServer

Writing MYMETA.yml and MYMETA.json

MNOONING/PlRPC/PlRPC-0.2020.tar.gz

C:\Perl64\bin\perl.exe Makefile.PL INSTALLDIRS=site -- OK

Running make for M/MN/MNOONING/PlRPC/PlRPC-0.2020.tar.gz

---- Unsatisfied dependencies detected during ----

---- MNOONING/PlRPC/PlRPC-0.2020.tar.gz ----

Net::Daemon [requires]

Running install for module 'Net::Daemon'

MNOONING/Net-Daemon-0.48.tar.gz

Has already been unwrapped into directory C:\Perl64\cpan\build\Net-Daemon-0.48-eudQww

MNOONING/Net-Daemon-0.48.tar.gz

Has already been prepared

MNOONING/Net-Daemon-0.48.tar.gz

Has already been made

MNOONING/Net-Daemon-0.48.tar.gz

Won't repeat unsuccessful test during this command

MNOONING/PlRPC/PlRPC-0.2020.tar.gz

Has already been unwrapped into directory C:\Perl64\cpan\build\PlRPC-8s9fqB

MNOONING/PlRPC/PlRPC-0.2020.tar.gz

Has already been prepared

Running make for M/MN/MNOONING/PlRPC/PlRPC-0.2020.tar.gz

Warning: Prerequisite 'Net::Daemon => 0.13' for 'MNOONING/PlRPC/PlRPC-0.2020.tar.gz' failed when processing 'MNOONING/Net-Daemon-0.48.ta

r.gz' with 'make\_test => NO'. Continuing, but chances to succeed are limited.

perldoc -t lib/RPC/PlServer.pm >README

cp lib/RPC/PlServer/Test.pm blib\lib\RPC\PlServer\Test.pm

cp lib/RPC/PlClient/Comm.pm blib\lib\RPC\PlClient\Comm.pm

cp lib/RPC/PlServer/Comm.pm blib\lib\RPC\PlServer\Comm.pm

cp lib/RPC/PlClient.pm blib\lib\RPC\PlClient.pm

cp lib/RPC/PlServer.pm blib\lib\RPC\PlServer.pm

cp lib/Bundle/PlRPC.pm blib\lib\Bundle\PlRPC.pm

MNOONING/PlRPC/PlRPC-0.2020.tar.gz

C:\Perl64\site\bin\dmake.exe -- OK

Running make test

C:\Perl64\bin\perl.exe "-MExtUtils::Command::MM" "-MTest::Harness" "-e" "undef \*Test::Harness::Switches; test\_harness(0, 'blib\lib', 'bl

ib\arch')" t/\*.t

t/base.t ...... Can't locate Net/Daemon.pm in @INC (you may need to install the Net::Daemon module) (@INC contains: C:\Perl64\cpan\build

\PlRPC-8s9fqB\blib\lib C:\Perl64\cpan\build\PlRPC-8s9fqB\blib\arch C:/Perl64/site/lib C:/Perl64/lib .) at C:\Perl64\cpan\build\PlRPC-8s9

fqB\blib\lib/RPC/PlServer.pm line 20.

Compilation failed in require at t/base.t line 7.

t/base.t ...... Dubious, test returned 2 (wstat 512, 0x200)

Failed 2/3 subtests

t/client.t .... Can't locate Net/Daemon/Test.pm in @INC (you may need to install the Net::Daemon::Test module) (@INC contains: C:\Perl64

\cpan\build\PlRPC-8s9fqB\blib\lib C:\Perl64\cpan\build\PlRPC-8s9fqB\blib\arch C:/Perl64/site/lib C:/Perl64/lib .) at t/lib.pl line 3.

Compilation failed in require at t/client.t line 7.

t/client.t .... Dubious, test returned 2 (wstat 512, 0x200)

No subtests run

t/compress.t .. Can't locate Net/Daemon/Test.pm in @INC (you may need to install the Net::Daemon::Test module) (@INC contains: C:\Perl64

\cpan\build\PlRPC-8s9fqB\blib\lib C:\Perl64\cpan\build\PlRPC-8s9fqB\blib\arch C:/Perl64/site/lib C:/Perl64/lib .) at t/lib.pl line 3.

Compilation failed in require at t/compress.t line 13.

t/compress.t .. Dubious, test returned 2 (wstat 512, 0x200)

No subtests run

t/crypt.t ..... skipped: (no reason given)

t/methods.t ... Can't locate Net/Daemon/Test.pm in @INC (you may need to install the Net::Daemon::Test module) (@INC contains: C:\Perl64

\cpan\build\PlRPC-8s9fqB\blib\lib C:\Perl64\cpan\build\PlRPC-8s9fqB\blib\arch C:/Perl64/site/lib C:/Perl64/lib .) at t/lib.pl line 3.

Compilation failed in require at t/methods.t line 7.

t/methods.t ... Dubious, test returned 2 (wstat 512, 0x200)

No subtests run

Test Summary Report

-------------------

t/base.t (Wstat: 512 Tests: 1 Failed: 0)

Non-zero exit status: 2

Parse errors: Bad plan. You planned 3 tests but ran 1.

t/client.t (Wstat: 512 Tests: 0 Failed: 0)

Non-zero exit status: 2

Parse errors: No plan found in TAP output

t/compress.t (Wstat: 512 Tests: 0 Failed: 0)

Non-zero exit status: 2

Parse errors: No plan found in TAP output

t/methods.t (Wstat: 512 Tests: 0 Failed: 0)

Non-zero exit status: 2

Parse errors: No plan found in TAP output

Files=5, Tests=1, 1 wallclock secs ( 0.03 usr + 0.03 sys = 0.06 CPU)

Result: FAIL

Failed 4/5 test programs. 0/1 subtests failed.

dmake.exe: Error code 130, while making 'test\_dynamic'

MNOONING/PlRPC/PlRPC-0.2020.tar.gz

one dependency not OK (Net::Daemon); additionally test harness failed

C:\Perl64\site\bin\dmake.exe test -- NOT OK

//hint// to see the cpan-testers results for installing this module, try:

reports MNOONING/PlRPC/PlRPC-0.2020.tar.gz

Running install for module 'DBD::Multiplex'

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/T/TK/TKISHEL/DBD-Multiplex-2.11.tar.gz

Fetching with LWP:

http://ppm.activestate.com/CPAN/authors/id/T/TK/TKISHEL/CHECKSUMS

Checksum for C:\Perl64\cpan\sources\authors\id\T\TK\TKISHEL\DBD-Multiplex-2.11.tar.gz ok

DBD-Multiplex/

DBD-Multiplex/Changes

DBD-Multiplex/lib/

DBD-Multiplex/Makefile.PL

DBD-Multiplex/MANIFEST

DBD-Multiplex/META.yml

DBD-Multiplex/other/

DBD-Multiplex/README

DBD-Multiplex/t/

DBD-Multiplex/t/test.t

DBD-Multiplex/other/example.pl

DBD-Multiplex/other/test.pl

DBD-Multiplex/lib/DBD/

DBD-Multiplex/lib/DBD/Multiplex.pm

Configuring T/TK/TKISHEL/DBD-Multiplex-2.11.tar.gz with Makefile.PL

Checking if your kit is complete...

Looks good

Generating a dmake-style Makefile

Writing Makefile for DBD::Multiplex

Writing MYMETA.yml and MYMETA.json

TKISHEL/DBD-Multiplex-2.11.tar.gz

C:\Perl64\bin\perl.exe Makefile.PL INSTALLDIRS=site -- OK

Running make for T/TK/TKISHEL/DBD-Multiplex-2.11.tar.gz

cp lib/DBD/Multiplex.pm blib\lib\DBD\Multiplex.pm

TKISHEL/DBD-Multiplex-2.11.tar.gz

C:\Perl64\site\bin\dmake.exe -- OK

Running make test

C:\Perl64\bin\perl.exe "-MExtUtils::Command::MM" "-MTest::Harness" "-e" "undef \*Test::Harness::Switches; test\_harness(0, 'blib\lib', 'bl

ib\arch')" t/\*.t

t/test.t .. ok

All tests successful.

Files=1, Tests=1, 0 wallclock secs ( 0.05 usr + 0.03 sys = 0.08 CPU)

Result: PASS

TKISHEL/DBD-Multiplex-2.11.tar.gz

C:\Perl64\site\bin\dmake.exe test -- OK

Running make install

Prepending C:\Perl64\cpan\build\DBD-Multiplex-GoF0hm/blib/arch C:\Perl64\cpan\build\DBD-Multiplex-GoF0hm/blib/lib to PERL5LIB for 'insta

ll'

Installing C:\Perl64\html\site\lib\DBD\Multiplex.html

Installing C:\Perl64\site\lib\DBD\Multiplex.pm

Appending installation info to C:\Perl64\lib/perllocal.pod

TKISHEL/DBD-Multiplex-2.11.tar.gz

C:\Perl64\site\bin\dmake.exe install -- OK

C:\Perl64\bin>

When I ran thee install on the desktop, it returned an error:

“It looks like you don’t have a C compiler and make utility installed.   
Trying to install dmake and the MinGW gcc compile using the   
Perl Package Manager (PPM). This may take a few minutes.”  
… “Please use the dmake to use the makefile.”…

For the DBD, per instructions at <http://search.cpan.org/~capttofu/DBD-mysql-4.029/lib/DBD/mysql/INSTALL.pod#ActiveState_Perl>

### ActiveState Perl

ActivePerl offers a PPM archive of DBD::mysql. All you need to do is typing in a cmd.exe window:

ppm install DBD-mysql

Results:

C:\Perl64\bin>ppm install DBD-mysql

Downloading ActiveState Package Repository packlist...done

Downloading log4perl packlist...redirect

Downloading log4perl packlist...done

Downloading log4perl Log-Dispatch-FileRotate PPD...not modified

Downloading log4perl Log-Dispatch PPD...not modified

Downloading log4perl Log-Log4perl PPD...not modified

Downloading log4perl Params-Validate PPD...not modified

Syncing site PPM database with .packlists...done

Syncing perl PPM database with .packlists...done

Downloading DBD-mysql-4.029...done

Unpacking DBD-mysql-4.029...done

Generating HTML for DBD-mysql-4.029...done

Updating files in site area...done

11 files installed

C:\Perl64\bin>

8TH Class – Wednesday evening, February 25, 2015

Perl is OOP.

Build.major.minor

1. – 1987
2. – 1988
3. – 1991
4. – 1992
5. – 1994
6. .

Creating classes from objects.

In Java, but a collection of classes is called a package.

In .NET, we have a namespace that contains classes.

In Python, we have a Module which contains a series of classes, modules, functions, and properties.

The last line in the module has to return 1.

In Perl, we have Packages for classes.

#!/usr/bin/perl #usually

But with packages we start:

package Calculator

….. we created Calculator.pm

‘HTML-Template’ module for using a templat factory.

Mustache – handlebars, Google Angular,

ad62160,

Command line parameters in Perl

* perl assignment3.pl level 55

\*!/usr/bin/perl -w

#@ARGV array has a size of 2

With other languages the index of 0 is the script name

With Perl languages the index of 0 is the script name

There is a special varia

$\ newline delimiter

$\_ scalar

@\_ calling a subroutine

$. File IO

$! Error upon dieing

Tonight:

$0 is the script name

# good because it will always work even if the script name is changed.

$argc = scalar(@ARGV);

If ($argc! = 2) {

Print “Usage: $0 param1 param2”;

Exit;

}

$word = $ARGV[0];

# Use regex:

# Trim the input.

# Then eliminate white space by using regex:

$word = ~ s/^\s+|\s+$//;

OOP

Call a module in Perl in multiple files: