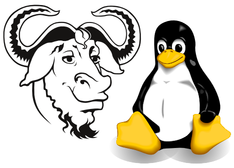

An Introduction to Perl — Solutions

Log into your VTC student email account. I have sent you an email (using a Perl program, of course!) that tells you your password for our laboratory computer system. Please read this email, then log into your Linux account.

Open the editor of your choice; available options are:
`emacs`, `xemacs`, `vi`, `gvim`, `gnp`, `pico`, `gedit`, and many others.

How to read about built in Perl functions: There is a huge amount of documentation about Perl installed on your computer. The Perl built-in functions are all described in detail in the `perlfunc` man page. But there is a handy tool for reading about a single built-in function:

`perldoc -f` *`function`*
This works in Windows as well as Linux.
The man page for perl operators is called `perlop`.

1. Write a program that calculates the circumference of a circle with a radius of 12.5. The circumference is $C = 2\pi r$, and $\pi \approx 3.1415927$.

```
#! /usr/bin/perl -w

use strict;
our $pi = 3.1415927;
our $r = 12.5;
our $c = 2 * $pi * $r;

print "Circumference of a circle with radius $r is $c\n";
```

2. Modify the program you just wrote to prompt for and read the radius from the person who runs the program.

```
#! /usr/bin/perl -w

use strict;
our $pi = 3.1415927;
print "Enter radius: ";
our $r = <STDIN>;
chomp $r;
our $c = 2 * $pi * $r;

print "Circumference of a circle with radius $r is $c\n";
```

3. Write a program that prompts for and reads two numbers, and prints out the result of multiplying the two numbers together.

```
#! /usr/bin/perl -w

use strict;
print "Enter first number: ";
our $n1 = <STDIN>;
chomp $n1;
print "Enter second number: ";
our $n2 = <STDIN>;
chomp $n2;
our $product = $n1 * $n2;

print "Product of $n1 and $n2 is $product\n";
```

4. Write a program that prompts for and reads a string and a number, and prints the string the number of times indicated by the number, on separate lines. Hint: read about the “x” operator.

```
#! /usr/bin/perl -w

use strict;
print "Enter string: ";
our $string = <STDIN>;
print "Enter number of times to repeat string: ";
our $n = <STDIN>;
chomp $n;
our $result = $string x $n;

print "$result";
```

5. Write a program that works like `cat`, but reverses the order of the lines. It should read the lines from files given on the command line, or from standard input if no files are given on the command line. Hint: read about the built-in `reverse` function. You will want to use an array.

Here is one possible solution:

```
#! /usr/bin/perl -w

use strict;
our @lines = <>;
print reverse @lines;
```

Here is another:

```
#! /usr/bin/perl -w

use strict;
# See perldoc -f reverse:
print reverse <>;
```