

Week 08 Laboratory Exercises

Objectives

- Developing Perl & Shell skills
- Exploring simple approaches to scraping data from the web

Preparation

Before the lab you should re-read the relevant lecture slides and their accompanying examples.

Getting Started

Create a new directory for this lab called `lab08`, change to this directory, and fetch the provided code for this week by running these commands:

```
$ mkdir lab08
$ cd lab08
$ 2041 fetch lab08
```

Or, if you're not working on CSE, you can download the provided code as a [zip file](#) or a [tar file](#).

EXERCISE:

What Courses Does UNSW Have this Year - Shell

Write a Shell script `courses.sh` which given a course prefix, e.g. COMP, prints all the course code and name for all UNSW shell_courses with that prefix offered this year on the Kensington Campus. For example:

```
$ ./courses.sh VISM
VISN1101 Seeing the World: Perspectives from Vision Science
VISN1111 Geometrical and Physical Optics
VISN1221 Visual Optics
VISN2111 Ocular Anatomy and Physiology
VISN2211 Organisation and Function of the Visual System
VISN3111 Development and Aging of the Visual System
VISN4016 Vision Science Honours
$ ./courses.sh COMP|tail
COMP9511 Human Computer Interaction
COMP9517 Computer Vision
COMP9596 Research Project
COMP9801 Extended Design and Analysis of Algorithms
COMP9814 Extended Artificial Intelligence
COMP9844 Extended Neural Networks and Deep Learning
COMP9900 Information Technology Project
COMP9991 Research Project A
COMP9992 Research Project B
COMP9993 Research Project C
```

Your program must be POSIX compatible Shell. You can assume anything that works with the version of `/bin/dash` on CSE systems is POSIX compatible.

You are not permitted to use other languages such as Perl, Python, C, ...

You are permitted to use programs such as `egrep`, `sed`, `sort`, `uniq`.

Hints:

The information you need for course code prefix COMP can be found in this web page:

<http://www.timetable.unsw.edu.au/current/COMPKENS.html>. You can assume this is the case for all prefixes.

The command `curl` will download a URL and print it to standard output.

```
$ curl --location --silent http://www.timetable.unsw.edu.au/current/COMPKENS.html | head
<title>Class Search by Teaching Period</title>
<link rel="stylesheet" type="text/css" href="../layout/2020/myunsw.css">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

  <table width="100%" cellpadding="0" cellspacing="0">
    <form name="googleForm" method="GET" action="http://www.google.com/u/theuniversityofnewsouthwales"
target="_blank">
      <tr>
        <td width="30%" style="height:120px; border-bottom:10px solid #FFCC00; background-color:#fff;"><a
href="http://www.unsw.edu.au" target="_blank">
</a></td>
        <td width="70%" style="height:120px; border-bottom:10px solid #FFCC00; background-color:#fff;
vertical-align:bottom; " align="right">
$
```

In a script it is best run as `curl --silent` so it doesn't print extra information on standard error.

The `--location` is required so `curl` will follow a HTTP redirect from the URL.

The `wget -q -O-` could also be used.

You may find `uniq's -w` option useful when removing duplicate courses.

When you think your program is working, you can use `autotest` to run some simple automated tests:

```
$ 2041 autotest shell_courses
```

When you are finished working on this exercise, you must submit your work by running `give`:

```
$ give cs2041 lab08_shell_courses courses.sh
```

before **Tuesday 28 July 18:00** to obtain the marks for this lab exercise.

EXERCISE:

What Courses Does UNSW Have this Year - Perl

Write a Perl script `courses.pl` which given a course prefix, e.g. COMP, prints all the course code and name for all UNSW courses with that prefix offered this year on the Kensington Campus. For example:

```
./courses.pl VISN
VISN1101 Seeing the World: Perspectives from Vision Science
VISN1111 Geometrical and Physical Optics
VISN1221 Visual Optics
VISN2111 Ocular Anatomy and Physiology
VISN2211 Organisation and Function of the Visual System
VISN3111 Development and Aging of the Visual System
VISN4016 Vision Science Honours
./courses.pl COMP|tail
COMP9511 Human Computer Interaction
COMP9517 Computer Vision
COMP9596 Research Project
COMP9801 Extended Design and Analysis of Algorithms
COMP9814 Extended Artificial Intelligence
COMP9844 Extended Neural Networks and Deep Learning
COMP9900 Information Technology Project
COMP9991 Research Project A
COMP9992 Research Project B
COMP9993 Research Project C
```

Your answer must be Perl only.

You may not run external programs, e.g. via `system` or backquotes.

Hints:

The information you need for course code prefix COMP can be found in this web page:
<http://www.timetable.unsw.edu.au/current/COMPKENS.html>. You can assume this is the case for all prefixes.

The Perl module **LWP::Simple** provides a very simple way to get a web page, e.g:

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

use LWP::Simple;
$url = "http://www.timetable.unsw.edu.au/current/COMPKENS.html";
$web_page = get($url) or die "Unable to get $url";
print $web_page;
```

You will need to remove duplicate entries for some courses.

When you think your program is working, you can use autotest to run some simple automated tests:

```
$ 2041 autotest perl_courses
```

When you are finished working on this exercise, you must submit your work by running give:

```
$ give cs2041 lab08_perl_courses courses.pl
```

before **Tuesday 28 July 18:00** to obtain the marks for this lab exercise.

CHALLENGE EXERCISE:

When Do These Courses Have Lectures

Write a Perl script `timetable.pl` which given course codes prints an ASCII timetable of their lecture times for this year in the format shown in the example below.

```
$ ./timetable.pl COMP2041 MATH2400 MATH2859
      Mon   Tue   Wed   Thu   Fri
09:00    L             L
10:00    L             L
11:00         L       L   L
12:00         L
13:00
14:00         L
15:00         L
16:00
17:00
18:00
19:00
20:00
$ ./timetable.pl COMP9044 COMP6771 GS0E9820
      Mon   Tue   Wed   Thu   Fri
09:00
10:00                L
11:00                L
12:00
13:00           L
14:00         L   L
15:00         L
16:00                L
17:00                L
18:00                L
19:00                L
20:00
```

You can assume that a course's lecture times will be found in a web page equivalent to:

<http://timetable.unsw.edu.au/current/MATH1131.html>.

Hint: if you find it difficult to use a regex to extract the line containing the lecture description, split the page into line match a previous line, then skip a certain number of lines.

You assume there are no lectures on weekends, and no lectures before 09:00 or after 21:00

Your answer must be Perl only.

You may not run external programs, e.g. via `system` or backquotes.

If a course is running in multiple terms print all the terms.

If there are multiple streams print all the streams, for example:

```
$ ./timetable.pl COMP1511
      Mon   Tue   Wed   Thu   Fri
09:00           L
10:00           L
11:00             L
12:00             L
13:00
14:00
15:00
16:00             L
17:00             L
18:00
19:00
20:00
$ ./timetable.pl MATH1131
      Mon   Tue   Wed   Thu   Fri
09:00           L
10:00           L
11:00
12:00             L
13:00      L     L     L
14:00      L     L             L     L
15:00      L     L             L     L
16:00      L             L     L
17:00           L
18:00
19:00
20:00
```

Note the correct output in all the above examples will change when next term's lectures are added to the timetable.

When you think your program is working, you can use autotest to run some simple automated tests:

```
$ 2041 autotest timetable
```

When you are finished working on this exercise, you must submit your work by running give:

```
$ give cs2041 lab08_timetable timetable.pl
```

before **Tuesday 28 July 18:00** to obtain the marks for this lab exercise.

Submission

When you are finished each exercises make sure you submit your work by running give.

You can run give multiple times. Only your last submission will be marked.

Don't submit any exercises you haven't attempted.

If you are working at home, you may find it more convenient to upload your work via [give's web interface](#).

Remember you have until **Tuesday 28 July 18:00** to submit your work.

You cannot obtain marks by e-mailing your code to tutors or lecturers.

You check the files you have submitted [here](#).

Automarking will be run by the lecturer several days after the submission deadline, using test cases different to those autotest runs for you. (Hint: do your own testing as well as running autotest.)

After automarking is run by the lecturer you can [view your results here](#). The resulting mark will also be available [via give's web interface](#).

Lab Marks

When all components of a lab are automarked you should be able to view the the marks [via give's web interface](#) or by running this command on a CSE machine:

```
$ 2041 classrun -sturec
```

COMP(2041|9044) 20T2: Software Construction is brought to you by
the [School of Computer Science and Engineering](#)
at the [University of New South Wales](#), Sydney.

For all enquiries, please email the class account at cs2041@cse.unsw.edu.au

CRICOS Provider 00098G