Sample Answers

1. A student wrote this script named **start_lab04.sh** to run before the Week 4 lab.

```
#!/bin/sh
cd ~/labs/04
ex1=jpg2png
ex2=email_image
ex3=date_image
ex4=tag_music
```

But when he ran his script, it didn't seem to work:

```
$ pwd
/home/z1234567
$ ./start_lab04.sh
$ pwd
/home/z1234567
$ echo $ex1 $ex2 $ex3 $ex4

```

Why not, and can you fix the shell script?

A shell script is executed by a separate shell so changes to its working directory affect only it. Similarly changes to variables in it, affect only it.
You can indicate that the commands in a file are to be run by a shell rather than executed as a separate program like this.

```
$ . ./start_lab04.sh
$ pwd
/home/z1234567/labs/04
$ echo $ex1 $ex2 $ex3 $ex4
jpg2png email_image date_image tag_music
```

2. The course code for COMP2041 has been changed to COMP2042 and the course code for COMP9041 has been changed to COMP9042. Write a shell script, `update_course_code.sh` which appropriately changes the course_code in all the files it is given as argument.

Sample solution for update_course_code.sh

```
#!/bin/sh

for file in "$@"
do
    temporary_file="$file.tmp.$$"
    if test -e "$temporary_file"
    then
        echo "$temporary_file" already exists
        exit 1
    fi
    sed 's/COMP2041/COMP2042/g;s/COMP9041/COMP9042/g' $file >$temporary_file &&
    mv $temporary_file $file
done
```

Alternatively a single line solution using sed's -i (--inplace) option which widely but not universely supported. Perl has a similar option:

```
$ sed -i 's/COMP2041/COMP2042/g;s/COMP9041/COMP9042/g' file1 file2 ...
```

3. Modify `update_course_code.sh` so if given a directory as argument it updates the course codes in files found in that directory and its sub-directories.

Sample solution for update_course_code.sh

```
#!/bin/sh

# doesn't handle pathnames containing white space
for file in `find "$@" -type f`
do
    temporary_file="$file.tmp.$$"
    if test -e "$temporary_file"
    then
        echo "$temporary_file" already exists
        exit 1
    fi
    sed 's/COMP2041/COMP2042/g;s/COMP9041/COMP9042/g' "$file" >"$temporary_file" &&
    mv -- "$temporary_file" "$file"
done
```

Sample solution for update_course_code.sh

```
#!/bin/sh
# doesn't hande pathnames containing new lines
find "$@" -type f|
while read file
do
    temporary_file="$file.tmp.$$"
    if test -e "$temporary_file"
    then
        echo "$temporary_file" already exists
        exit 1
    fi
    sed 's/COMP2041/COMP2042/g;s/COMP9041/COMP9042/g' "$file" >"$temporary_file" &&
    # not use of -- to stop mv interpreting filename beginning with - as option
    mv -- "$temporary_file" "$file"
done
```

Recursive sample solution for update_course_code.sh

```
#!/bin/sh

# solution by alex linker 2018-08-17

for arg in "$@"
do
    if [ -d $arg ] #if it's a directory
    then
        $0 "$arg"/* # call self with everything in the given directory
    else
        sed -i -E 's/(COMP[29]04)1/\12/g;' "$arg" # uses capture groups to replace the last char
        # note that sed's -i option does it inplace but this option is not universally supported
    fi
done
```

4. Write a shell script, `is_business_hours` which exits with a status of 0 if the current time is between 9am & 5pm, and otherwise exits with a status of 1.
   Hint: the date command prints the current time in a format like this:

```
$ date
Sun Mar 18 12:57:08 EST 2012
```

Sample solution for is_business_hours.sh

```
#!/bin/sh
# exits with a status of 0  if the current time is between 9am & 5pm
# otherwise exit with a status 1
#
# date format looks like this Sun Mar 18 12:57:08 EST 2012

current_hour=`date|cut -d\  -f4|cut -d: -f1`
if test $current_hour -ge 9 -a $current_hour -lt 17
then
    exit 0
else
    exit 1
fi
```

Another sample solution for is_business_hours.sh

```
#!/bin/sh
# exits with a status of 0  if the current time is between 9am & 5pm
# otherwise exit with a status 1
# date output looks like this 'Sun Mar 18 12:57:08 EST 2012'
# relies on the exit status being the exit status of last command
# when there isn't an explicit exit

current_hour=`date|cut -d\  -f4|cut -d: -f1`
test $current_hour -ge 9 -a $current_hour -lt 17
```

5. CSE systems have a command, `mlalias`, which prints information about a specified mail alias. For example:

```
$ mlalias COMP2041-list
        alias: COMP2041-list
  description: Udb alias list
    addresses:
                z5000000
                z5000001
                .......
                z5555555
                andrewt
       owners: udb, cs2041
authorised posters: @Employee, @Subject_Utility, @Wheel
    Moderator: udb
       Status: system, closed, moderated, virtual, and public
```

Convert the output of the mlalias command into a new line separated list of CSE usernames, like this:

```
z5000000
z5000001
.......
z5555555
andrewt
```

```
mlalias COMP2041-list|egrep -v :|sed 's/^ *//'
```

6. CSE system have a command, `acc`, which prints information about a specified user. For example:

```
$ acc z5555555
            User Name : z5555555                Aliases :
                  Uid : 25068
               Groups :
              Expires : 31 Dec 2018
         User classes : 3978_Student, COMP2041_Student[15dec2018]
                      : COMP2121_Student[15dec2018], COMP2511_Stud
ent[15dec2018]
                      : COMP1511_Tutor[16dec2018], COMP3900_Studen
t[15dec2018]
                 Name : Michael Yang Zhou
Password last changed : 2018/03/02.21:23:19
       Home Directory : /import/adams/1/mzhou
                 Name : Mr Zhou, Michael Yang (Michael Yang Zhou)
             Position : UGRD (Faculty of Engineering)
         UNSW Number : 5555555
           UNSW Mail : z5555555@unsw.edu.au
           UNSW Home : //INFPWFS219.ad.unsw.edu.au/Student038$/z55
55555
            CSE Home : /import/kamen/3/z5555555
```

Write a pipeline which converts the output of acc into a new line separated list of courses the person is enrolled in, like this:

```
COMP2041
COMP2121
COMP2511
COMP3900
```

Make sure you don't include COMP1511 which Michael tutors.

```
$ acc z5555555 |cut -d: -f2
 z5555555              Aliases
 25068

 31 Dec 2018
 3978_Student, COMP2041_Student[15dec2018]
 COMP2121_Student[15dec2018], COMP2511_Student[15dec2018]
 COMP1511_Tutor[16dec2018], COMP3900_Student[15dec2018]
 Michael Yang Zhou
 2018/03/02.21
 /import/adams/1/mzhou
 Mr Zhou, Michael Yang (Michael Yang Zhou)
 UGRD (Faculty of Engineering)
 5555555
 z5555555@unsw.edu.au
 //INFPWFS219.ad.unsw.edu.au/Student038$/z5555555
 /import/kamen/3/z5555555
$ acc z5555555 |cut -d: -f2|tr , '\n'
 z5555555              Aliases
 25068

 31 Dec 2018
 3978_Student
 COMP2041_Student[15dec2018]
 COMP2121_Student[15dec2018]
 COMP2511_Student[15dec2018]
 COMP1511_Tutor[16dec2018]
 COMP3900_Student[15dec2018]
 Michael Yang Zhou
 2018/03/02.21
 /import/adams/1/mzhou
 Mr Zhou
 Michael Yang (Michael Yang Zhou)
 UGRD (Faculty of Engineering)
 5555555
 z5555555@unsw.edu.au
 //INFPWFS219.ad.unsw.edu.au/Student038$/z5555555
 /import/kamen/3/z5555555
$ acc z5555555 |cut -d: -f2|tr , '\n'|egrep _Student
 3978_Student
 COMP2041_Student[15dec2018]
 COMP2121_Student[15dec2018]
 COMP2511_Student[15dec2018]
 COMP3900_Student[15dec2018]
$ acc z5555555 |cut -d: -f2|tr , '\n'|egrep _Student|cut -c2-9
3978_Stu
COMP2041

COMP2121
COMP2511
COMP3900
$ acc z5555555 |cut -d: -f2|tr , '\n'|egrep _Student|cut -c2-9|egrep '[A-Z][A-Z][A-Z][A-Z]
[0-9][0-9][0-9][0-9]'
COMP2041
COMP2121
COMP2511
COMP3900
$ acc z5555555 | tr , "\n" | sed -nE 's/^.*([A-Z]{4}[0-9]{4})_Student.*/\1/p' #Option 2

COMP2041
COMP2121
```

```
COMP2511
COMP3900
```

7. Use the pipeines from the above 2 questions to write shell commands which print a list of courses taken by COMP2041 students with counts of how many COMP2041 students take each, like this:

```
     55 COMP2911
     37 COMP2121
     17 COMP3311
     10 COMP2111
      9 COMP3331
      ..........
```

```
mlalias COMP2041-list|
egrep -v :|
sed 's/^ *//'|
while read zid
do
    acc $zid|
    cut -d: -f2|
    tr , '\n'|
    egrep _Student|
    cut -c2-9|
    egrep '[A-Z][A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9]'
done|
sort|
uniq -c|sort -rn
```

8. COMP2041 student Shruti has a 'friends' subdirectory in her home directory that contains images of her many friends. Shruti likes to view these images often and would like to have them appear in other directories within her CSE account so she has written a shell script to symbolically link them to the current directory:

```
for image_file in `ls ~/friends`
do
    ln -s "~/friends/$image_file" .
done
```

The links created by Shruti's script are broken. Why? How can she fix her script?

The shell does not replace tilde (~) with the user's home directory inside double-quotes, and does not handle spaces in filenames correctly. For example:

```
$ echo ~
/home/shruti
$ echo "~"
~
$ touch a\ b
$ for f in `ls`; do echo $f; done
a
b
```

This should work for Shruti:

```
for image_file in ~/friends/*
do
    ln -s "$image_file" .
done
```

9. Write a shell script named `isprime` which given an integer as argment, tests whether it is prime and prints a suitable message:

```
$ isprime 42
42 is not prime
$ isprime 113
113 is prime
```

Your script should exit with a non-zero exit status if its argument is not prime.

Write a second script named `primes` which uses the first script to print all primes less than a specified value, e.g:

```
$ primes 100
2
3
5
7
11
13
17
...
79
83
89
97
```

Another sample solution for isprime.sh

```
#!/bin/sh
# test whether the specified integer is prime

if test $# != 1
then
    echo "Usage: $0 <number>"
    exit 1
fi

n=$1

i=2
while test $i -lt $n
do
    if test `expr $n % $i` -eq 0
    then
        echo "$n is not prime"
        exit 1
    fi
    i=`expr $i + 1`
done
echo "$n is prime"
```

Yet another solution for isprime.sh

```
#!/bin/sh
# test whether the specified integer is prime
# written by Han Zhao

if test $# != 1
then
        echo "Usage: $0 <number>"
        exit 1
fi

n=$1
if [ $1 -eq 1 ]
then
        echo "$n is not prime"
        exit 1
fi

i=2
while test $(expr $i \* $i) -le $n
do
        if test `expr $n % $i` -eq 0
        then
                echo "$n is not prime"
                exit 1
        fi
        i=`expr $i + 1`
done
echo "$n is prime"
exit 0
```

Sample solution for primes.sh

```
#!/bin/sh
# print the prime numbers less than the specified argument

case $# in
1) ;;
*) echo "Usage: $0 <number>"; exit 1
esac
limit=$1

p=2
while test $p -lt $limit
do
    if isprime.sh $p >/dev/null
    then
        echo $p
    fi
    p=`expr $p + 1`
done
exit 0
```

Revision questions

The remaining tutorial questions are primarily intended for revision - either this week or later in session.
Your tutor may still choose to cover some of the questions time permitting.

10. COMP2041 student Big Bad Barry tries to impress a girl at a party by betting her she can't work out what this shell script:

```
#!/bin/sh
IFS=abc
echo "$*"
```

prints when run like this:

```
$ ./script.sh mount inside
```

What does the script print?

Will the girl go out with Big Bad Barry?

The script will do this:

```
$ ./script.sh mount inside
mountainside
```

This is because `IFS` is a special internal shell variable which indicates the argument separators. The first character from IFS is uses to separate the argument when expanding $*.

Big Bad Barry won't get a date. No one is impressed by knowledge of hacky&obscure shell features. Good programmers avoid quirky little-known language features.