

THE HONG KONG UNIVERSITY OF SCIENCE & TECHNOLOGY

# COMP2021: Unix and Script Programming

Midterm Exam Spring 2013

30 Apr 2013

***Solution***

*Student Name:* \_\_\_\_\_

*Student ID:* \_\_\_\_\_

*Lab Section: LA1A / LA1B*

**Instructions:**

Write your name, student ID, and lab section on this page.

Answer all questions in the space provided.

**For Grading Purposes Only:**

Problem 1 \_\_\_\_\_/15  
Problem 2 \_\_\_\_\_/10  
Problem 3 \_\_\_\_\_/5  
Problem 4 \_\_\_\_\_/10  
Problem 5 \_\_\_\_\_/6  
Problem 6 \_\_\_\_\_/10  
Problem 7 \_\_\_\_\_/10  
Problem 8 \_\_\_\_\_/8  
Problem 9 \_\_\_\_\_/7  
Problem 10 \_\_\_\_\_/4  
Problem 11 \_\_\_\_\_/5  
Problem 12 \_\_\_\_\_/10

***Total*** \_\_\_\_\_/100

## Question 1 Multiple Choice [15 marks]

**UNIX utilities: circle the UNIX command that completes the task correctly.**

a) Delete the file `textfile`: *Answer C*

A. `del textfile`

C. `rm textfile`

B. `rmdir textfile`

D. `mv textfile`

b) Display the last 6 lines in file `textfile`: *Answer A*

A. `tail -6 textfile`

C. `more +6 textfile`

B. `more -6 textfile`

D. `tail +6 textfile`

c) Rename a file from `text1` to `text2`: *Answer: C*

A. `ren text1 text2`

C. `mv text1 text2`

B. `cat text1 > text2`

D. `mv text1 > text2`

d) List all the files in home directory and its subdirectories recursively: *Answer C*

A. `ls`

C. `ls -R`

B. `ls -l`

D. `ls -s`

e) Output the current date to a file named `date.txt`: *Answer A*

A. `date > date.txt`

C. `date >> date.txt`

B. `date | date.txt`

D. `date date.txt`

f) Display the lines in the man page of command "grep" containing the word "word": *Answer D*

A. `man grep > grep word`

C. `grep word < man grep`

B. `grep word | man grep`

D. `man grep | grep word`

g) Change the file content of `textfile` from view A (original) to view B (on screen): *Answer A*

View A:

```
bell
apple
bell
apple
```

View B:

```
bell
apple
```

A. `uniq textfile`

C. `unique textfile`

B. `cat textfile > uniq`

D. `rm textfile`

## General questions

h) Which statement will open the `/etc/passwd` file for reading only? **Answer B**

- A. `open(PASSFILE, "+>/etc/passwd");`
- B. `open(PASSFILE, "/etc/passwd");`
- C. `open(PASSFILE, "+</etc/passwd");`
- D. `open(PASSFILE, ">/etc/passwd");`

i) How many elements in `@array1` if it's defined as `@array1 = (9, "A", 0..9, "PERL")`?

**Answer A**

- A. 13
- B. 12
- C. 4
- D. 16

j) What is the output of the following code segment? (2 marks) **Answer D**

```
@array = ("Y", "W", "X");  
@array = sort(@array);  
unshift(@array, "Z");  
print($array[0]);
```

- A. W
- B. X
- C. Y
- D. Z

k) What is the output of the following code segment? (2 marks) **Answer C**

```
$_ = "COMP 2021 is fun";  
s/(\w+)/<$1>/g;  
print "$_\n";
```

- A. `<COMP> 2021 is fun`
- B. `COMP <2021> is fun`
- C. `<COMP> <2021> <is> <fun>`
- D. `<COMP> <is> <fun>`

l) What is the output of the following code segment? (2 marks) **Answer D**

```
%hashA = ("alpha", "beta", "gamma", "delta");  
%hashA = reverse(%hashA);  
Print $hashA{"alpha"};
```

- A. beta
- B. delta
- C. gamma
- D. The code fails execution

## Question 2: Unix file system and links [10 marks]

Read the following screen output and answer the questions.

```
$ ls -li
total 140
273908807 -rw-r--r-- 1 cindy comp2021 128079 Apr 8 11:32 exam
273908808 -rw-r--r-- 1 cindy comp2021      15 Apr 8 11:33 lab1
273908809 -rw-r--r-- 2 cindy comp2021    202 Apr 8 11:33 lab2
273908809 -rw-r--r-- 2 cindy comp2021    202 Apr 8 11:33 lab3
273908811 lrwxrwxrwx 1 cindy cs          4 Apr 8 11:36 lab4 -> lab3
403453165 drwxr-xr-x 2 cindy comp2021      6 Apr 8 11:31 lectures
542597928 drwxr-xr-x 2 cindy comp2021      6 Apr 8 11:37 docs
273908810 lrwxrwxrwx 1 cindy cs          8 Apr 8 11:35 slides -> lectures
```

a) How many directories are there? How many hard link(s) and symbolic link(s) are there? (2 marks)

*Answer: 2 directories, 3 links (or 4 links, or 8 links)*

b) Name all the actual files in the list. (1 mark)

*Answer: exam, lab1, lab2 (or lab3, but not both)*

c) What will happen if this command is issued: "rm lab3"? (2 marks)

*Answer: Only a link to lab3 is removed, so lab4 becomes a broken link. lab2 is not affected.*

d) The **comp2021** group includes all the COMP2021 staff (including **cindy**). What command should be issued so that file **exam** can be edited by all of them, while anyone else cannot see nor edit it? (2 marks)

*Answer: chmod 660 exam / chmod g+w,o-r exam (plus execute permission is okay)*

e) What command should be issued so that file **docs** is only accessible by **cindy** with full access right? (1 mark)

*Answer: chmod 700 docs / chmod go-rx docs*

f) Use a command to list out all the files, even for hidden files. (2 marks)

*Answer: ls -la*

### Question 3 Shell [5 marks]

Point out the error(s) in the following shell program and correct them.

```
Line 1:    #/bin/sh
Line 2:    echo "Enter height of parallelogram: "
Line 3:    read $height
Line 4:    echo "Enter width of parallelogram: "
Line 5:    read $width
Line 6:    area = 'expr $height * $width'
Line 7:    echo "The area of the parallelogram is $area"
```

Answers:

- 1) # /bin/sh should be #!/bin/sh
- 2) Delete "\$" in "read \$height" and "read \$width"
- 3) Need backslash before star in multiplication: expr \$height \\* \$width
- 4) Cannot have spaces in "area = ", should be: "area="
- 5) area = 'expr \$height \* \$width' should be with backquotes: area = `expr \$height \\* \$width`

## Question 4 Shell and Perl programming [10 marks]

You have done a “Big and Small” game in the lab. Translate this Perl program into a Shell program. Please write clearly. You may use a symbol ▽ to denote any necessary space.

Perl programming	Shell programming
<code>#!/usr/local/bin/perl</code>	<code>#!/bin/sh</code>
<code>srand;</code>	<code># no srand needed</code>
<code>while (1)</code>	<code>while [ 1 ] # (1 -- while loop)</code>
<code>{</code>	<code>Do</code>
<code>    \$dice = int(rand(6)) + 1;</code>	<code>    dice=`expr \$RANDOM % 6 + 1`</code>
<code>    if (\$dice &lt;= 3)</code>	<code>    if [ \$dice -le 3 ] # (1 -- cond.)</code>
<code>    {</code>	<code>    then</code>
<code>        \$result="S";</code>	<code>        result="S" # (.5)</code>
<code>    } else {</code>	<code>    else # (1 -- if)</code>
<code>        \$result="B";</code>	<code>        result="B" # (.5)</code>
<code>    }</code>	<code>    fi</code>
<code>    print "B or S?\n";</code>	<code>    echo "B or S?"</code>
<code>    chomp(\$input = &lt;STDIN&gt;);</code>	<code>    read input # (1)</code>
<code>    if (\$input ne "B" &amp;&amp; \$input ne "S")</code>	<code>    if [ \$input != "B" -a \$input != "S" ] # (1 -- cond.)</code>
<code>    {</code>	<code>    then</code>
<code>        last;</code>	<code>        break # (1)</code>
<code>    }</code>	<code>    fi</code>
<code>    print "* Result is \$result and your guess is \$input\n";</code>	<code>    echo "* Result is \$result and your guess is \$input" # (1)</code>
<code>    if ( \$input eq \$result )</code>	<code>    if [ \$input = \$result ] # (1)</code>
<code>    {</code>	<code>    then</code>
<code>        print "You have won!\n";</code>	<code>        echo "You have won!"</code>
<code>    } else {</code>	<code>    else</code>
<code>        print "You have lost!\n";</code>	<code>        echo "You have lost!"</code>
<code>    }</code>	<code>    fi</code>
<code>}</code>	<code>done # (1)</code>

Errors: “spacing” -1, “\$” -1, “brackets” -1

## Question 5 Perl basics [6 marks]

Complete the following “padding zeros” program `Pad0.pl` to read in user-input integers, and pad zeros to each integer to fill up the total number of digits, and output it to the screen. If user inputs blank lines or a line of a number 0, print a line with asterisks (\*) to fill up the expected number of digits. Assume the total number of digits is an argument of the program.

A sample execution of the program is as follows:

```
$ Pad0.pl 6
```

```
15
```

```
000015
```

```
0
```

```
*****
```

```
#!/usr/local/bin/perl5 -w
```

```
$TotalDigits = $ARGV[0];
```

```
print "nozero = $TotalDigits\n";
```

```
while (<STDIN>)
```

```
{
```

```
    $Line = _____;
```

```
    _____;
```

```
    if ($Line) {
```

```
        $NoDigit = length(_____);
```

```
        $ZerosPad = _____ - $NoDigit + 1;
```

```
        if ($ZerosPad < 0) {
```

```
            $ZerosPad = 0;
```

```
        }
```

```
        print "0"_____."$Line"."\\n";
```

```
    }
```

```
    else {
```

```
        print "*"_____. "\\n";
```

```
    }
```

```
}
```

```
$_
```

```
Chomp($Line)
```

```
$_ (or $Line)
```

```
$TotalDigits
```

```
x $ZerosPad
```

```
x $TotalDigits
```

## Question 6 Perl control flow [10 marks]

Write a Perl program `AddNumbers.pl` that sums the numbers from 1 to a user-input positive integer number ( $>1$ ), and also sums the even numbers. You can assume the user always gives correct input. A sample execution of the program is as follows:

```
$ AddNumbers.pl
Enter the last number in the sum:
7
The sum of the numbers 1 to 7 is 28
The sum of the even numbers is 12
```

Answer:

```
1:  #!/usr/local/bin/perl
2:
3:  print ("Enter the last number in the sum:\n");
4:  $limit = <STDIN>;
5:  chop ($limit);
6:  $count = 1;
7:  $total = $eventotal = 0;
8:  for ($count = 1; $count <= $limit; $count++) {
9:      $total += $count;
10:     if ($count % 2 == 1) {
11:         # start the next iteration if the number is odd
12:         next;
13:     }
14:     $eventotal += $count;
15: }
16: print("The sum of the numbers 1 to $limit is $total\n");
17: print("The sum of the even numbers is $eventotal\n");
```



## Question 7 Perl file I/O [10 marks]

Finish the Perl program `Mimic_wc.pl` which mimics the behavior of 'wc' Unix utility. The program takes a filename from the command line, and prints out the number of lines and words contained in that file. ('word' here simply means sequences of non-whitespace).

Content of text file `file1`:

HKUST Vision

To be the leading university with significant international impact  
and strong local commitment

A sample execution of the program is as follows:

```
$ Mimic_wc.pl file1
Lines: 3, Words: 15
$ Mimic_wc.pl nonexistentfile
Can't open file!
```

```
#!/usr/local/bin/perl5 -w
#get file name from command line
```

```
$filename = _____;
$words = 0;
$lines = 0;
```

```
# open filehandle FH for read-only mode, exit if file doesn't exist
```

```
_____;
```

```
while (<FH){
    # count lines and words in file
```

```
}
Print "Lines: $lines, Words: $words\n";
```

Answer:

```
#!/usr/local/bin/perl5 -w
$filename = $ARGV[0];
```

```

$words = 0;

$lines = 0;

open (FH, $filename) or die ("Can't open file!\n");
while(<FH>){
    $lines++;

    @words = /\s+/g;

    $words += @words;
}

print "Lines: $lines, Words: $words\n";

```

### **Question 8 Perl functions [8 marks]**

Read the following Perl program.

```

1:  #!/usr/local/bin/perl
2:
3:  $inputline = <STDIN>;
4:  $inputline =~ s/^\s+|\s+$//g;
5:  @list = split (/ \s+/, $inputline);
6:  $result = rightcalc (0);
7:  print ("The result is $result.\n");
8:
9:  sub rightcalc {
10:      my ($index) = @_ ;
11:      my ($result, $operand1, $operand2);
12:
13:      if ($index+3 == @list) {
14:          $operand2 = $list[$index+2];
15:      } else {
16:          $operand2 = rightcalc($index+2);
17:      }
18:      $operand1 = $list[$index+1];
19:      if ($list[$index] eq "+") {
20:          $result = $operand1 + $operand2;
21:      } elsif ($list[$index] eq "*") {
22:          $result = $operand1 * $operand2;
23:      } elsif ($list[$index] eq "-") {
24:          $result = $operand1 - $operand2;
25:      } else {
26:          $result = $operand1 / $operand2;
27:      }
28:      return $result;

```

29: }

- a) Assume the user inputs -  $20 * 3 + 2 \ 1$ , what is the content of `@list` after Line 5? (2 marks)

*"-", "20", "\*", "3", "+", "2", "1"*

- b) Does scalar variable `$result` in Line 6 and Line 20 refer to the same value? Why? (1 mark)

*No, `$result` in Line 20 is declared to be a local variable to function `rightcalc()`.*

- c) What does Line 13 do? (1 marks)

*It checked whether base case of recursion is reached. That is, the array only consists of 3 elements (2 operands and 1 operator). `@list` here returns the array length.*

- d) What is the usage of `@_` in Line 10? (1 mark)

*It holds all arguments passed from caller for the duration of subroutine.*

- e) What is the output of the Perl program is user inputs -  $20 * 3 + 2 \ 1$ ? Briefly explain why. (3 marks)

*The program calculates  $20 - 3*(2+1) = 11$ , it accepts a list of numbers and operands that is to be evaluated from right to left.*

### Question 9 Regular expressions [7 marks]

- a) Given the string below, write the value of `$1` when the Perl pattern `/(\w+:\s*\d+\.\d+)/` is applied to the string. Write "no match" if the string does not match the pattern. (4 marks)

6:16pm up 5 days, 23:26, Users, load average: 0.37, 0.20, 0.1

*Answer: average: 0.37*

- b) Which of the following string(s) can match pattern `/(a|b)(.)([aeiou])\1/`? Circle all your answers. (3 marks)

*Answers: BD (1 mark for each correct answer)*

- A. a!a!
- B. b-ob\*
- C. \_6a\_
- D. b!ab
- E. a\_i\_
- F. !ae

G. None of above

### **Question 10 Perl directory access [4 marks]**

Finish the program called `di.pl` to change to a directory specified by the user (assume the user inputs absolute path), then list the names of all files that end with `.cpp` under that directory. If the directory change did not succeed, exit the program with a warning message "Cannot chdir". You should print the current working directory at the very beginning.

For example: assume `/home/comp2021` contains "`lab1.cpp`", "`lab2.cpp`", "`device`", the current directory is `/home`

`$di.pl`

`/home`

`Where to?`

`/home/comp2021`

`lab1.cpp`

`lab2.cpp`

`#!/usr/local/bin/perl5 -w`

\_\_\_\_\_ ; # print current directory

`print "Where to?\n";`

`chomp($newdir = <STDIN>);`

\_\_\_\_\_ || die "Cannot chdir.\n"; # change directory

`foreach (_____ ) # check every file`

`{`

`if(/(\w+).cpp/){`

`print _____; # print file name if end with .cpp`

`}`

`}`

`#!/usr/local/bin/perl5 -w`

`system("pwd"); # 1 mark`

`print "Where to?\n";`

`chomp($newdir = <STDIN>);`

`chdir($newdir) || die "Cannot chdir to $newdir.\n"; # 1mark`

`foreach (<*>) # 1mark`

`{`

`if(/(\w+).cpp/){`

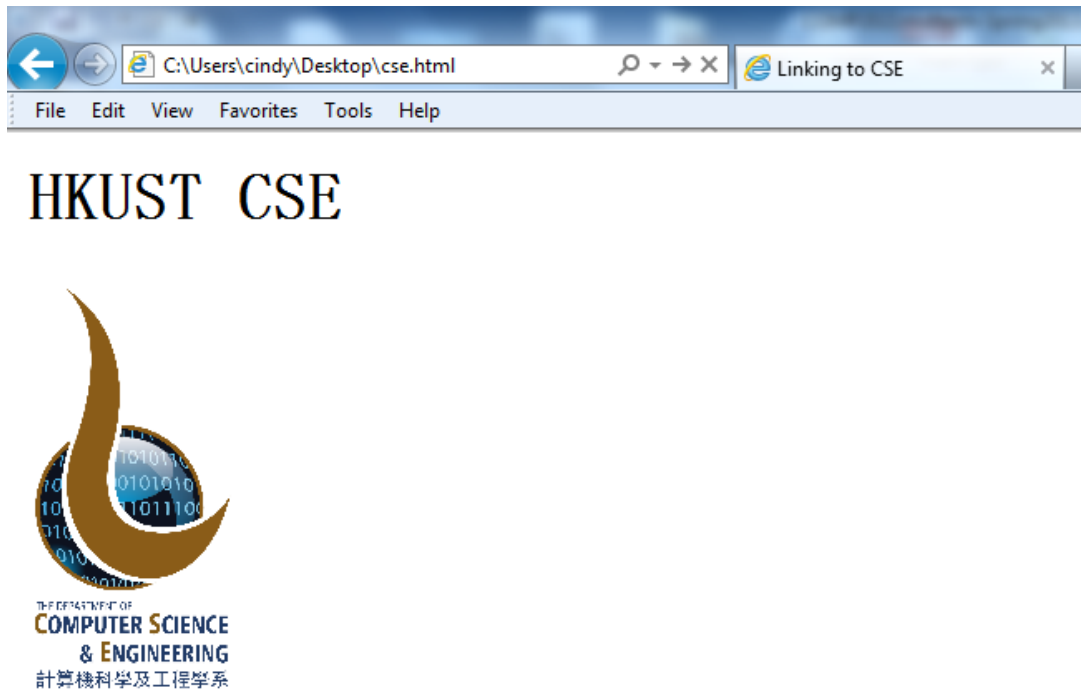
`print "$_\n";`

`}`

`}`

## Question 11 HTML [5 marks]

Mark and correct the errors in this HTML code so that the code is valid and it will be displayed as expected in any browser.



The CSE website is at <http://www.cse.ust.hk>

```
Line 1:    <html>
Line 2:    <head> Linking to CSE </head>
Line 3:    <body>
Line 4:    <h1>HKUST CSE</h1>
Line 5:    <img> src="logo.png" height="200" width="100" </img>
Line 6:    <p>The CSE website is at
Line 7:    <href="http://www.cse.ust.hk">http://www.cse.ust.hk</p>
Line 8:    </html>
```

Answer:

```
<html>
<head>
<title>Linking to CSE</title>
</head>
<body>
<h1>HKUST CSE</h1>

<p>The CSE website is at <a href="http://www.cse.ust.hk">http://www.cse.
ust.hk</a></p>
</body>
</html>
```

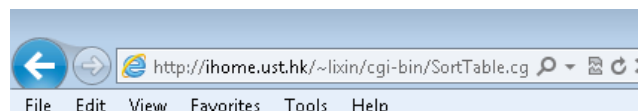
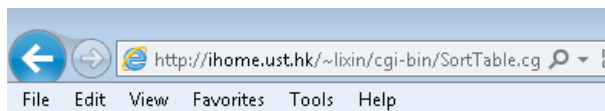
## Question 12 CGI and Hash [10 marks]

Fill in blanks in the following CGI program which displays student information (name and ID) stored in a text file named "data.txt". It should allow user to view the information in different sorting options (sort by name or ID, chosen by a popup menu and a submit button). Assume the text file is put in the same directory with your CGI program, and student's names and ID are unique.

### Content of data.txt

```
Peter Lee,02458934
Alice Wong,07235963
Bob Chan,05839201
Simon Kwok,08378495
Andrew Wong,09873510
Chris Tsui,06902312
Macro Lee,09872395
Samual Chan,04983861
```

### Sorting result using ID or Name



## Student Information

Sort Order

Name	ID
Peter Lee	02458934
Samual Chan	04983861
Bob Chan	05839201
Chris Tsui	06902312
Alice Wong	07235963
Simon Kwok	08378495
Macro Lee	09872395
Andrew Wong	09873510

## Student Information

Sort Order

Name	ID
Alice Wong	07235963
Andrew Wong	09873510
Bob Chan	05839201
Chris Tsui	06902312
Macro Lee	09872395
Peter Lee	02458934
Samual Chan	04983861
Simon Kwok	08378495

```

#!/usr/local/bin/perl

use _____;

open(DATA, "data.txt");

while (<DATA>) {
    chomp;
    ($name, $id) = split/,/;
    $student{$id} = $name;
}

print header;
print start_html("Student Information");
print h1("Student Information");

print _____;

print p("Sort Order",

    _____,

    _____);
print end_form;

print _____;
print "<table border='1'>\n";
print "<tr><td>Name</td><td>ID</td></tr>\n";

if (!param()) {
    foreach $id (keys %student) {
        print "<tr><td>$student{$id}</td><td>$id</td></tr>\n";
    }
} else {
    if (param("sort") eq "Name") {

        %studentrev = _____;

        foreach $name (_____) {

            print _____;

        }
    } else {
        foreach $id (_____) {
            print "<tr><td>$student{$id}</td><td>$id</td></tr>\n";
        }
    }
}

```

```
    }  
}  
    print _____;  
  
    print end_html;
```

*Answer:*

***CGI qw(:standard)***

***start\_form***

***popup\_menu("sort", ["Name", "ID"])***

***"Sort"***

***hr***

***reverse %student***

***sort keys %studentrev***

***"<tr><td>\$name</td><td>\$studentrev{\$name}</td></tr>\n"***

***sort keys %student***

***"</table>\n"***