Practical Computing for Scientists

Armin Sobhani

CSCI 2000U

UOIT – Fall 2015

UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

Midterm Exam – Part I

Question 1

a) What would be the current working directory at the end of the following command sequence?

```
$ pwd
/home/user1/proj
$ cd src
$ cd ../..
$ cd .
$ pwd
```

Question 1

a) What would be the current working directory at the end of the following command sequence?

```
$ pwd
/home/user1/proj
$ cd src
$ cd ../..
$ cd .
$ pwd
```

/home/user1

Question 1

b) What command(s) could you enter at the Unix command prompt to display a list of files with their corresponding sizes in human readable format?

Question 1

b) What command(s) could you enter at the Unix command prompt to display a list of files with their corresponding sizes in human readable format?

```
$ ls -s -h
```

Question 1

b) What command(s) could you enter at the Unix command prompt to display a list of files with their corresponding sizes in human readable format?

```
$ ls -s -h
$ ls -sh
```

Question 1

b) What command(s) could you enter at the Unix command prompt to display a list of files with their corresponding sizes in human readable format?

```
$ ls -s -h
$ ls -sh
$ ls -hs
```

Question 1

c) What command(s) could you enter at the Unix command prompt to create a new file '**`new.txt`**' that is a concatenation of '**`file1.txt`**' and '**`file2.txt`**'?

Question 1

c) What command(s) could you enter at the Unix command prompt to create a new file '**`new.txt`**' that is a concatenation of '**`file1.txt`**' and '**`file2.txt`**'?

```
$ cat file1.txt file2.txt
> new.txt
```

Question 1

d) What command(s) could you enter at the Unix command prompt to add group read/write permission to all the files in the current directory?

Question 1

d) What command(s) could you enter at the Unix command prompt to add group read/write permission to all the files in the current directory?

```
$ chmod g+rw *
```

Question 1

e) What command(s) could you enter at the Unix command prompt to find all the files in the current directory and sub-directories containing the word **`"sim"`** in their names?

Question 1

e) What command(s) could you enter at the Unix command prompt to find all the files in the current directory and sub-directories containing the word **"sim"** in their names?

```
$ find . -name '*sim*'
```

Question 1

f) What is the command to search all files in your current directory that contain the word **`"boson"`**?

Question 1

f) What is the command to search all files in your current directory that contain the word **"boson"**?

```
$ grep boson *
```

Question 1

g) What is the output of the following code?

```
$ os=Unix
$ echo 1.os 2.$os 3."$os" 4.'$os'
```

Question 1

g) What is the output of the following code?

`1.os 2.Unix 3.Unix 4.$os`

```
$ os=Unix
$ echo 1.os 2.$os 3."$os" 4.'$os'
```

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

a) What command(s) could you enter to stage the latest changes?

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

a) What command(s) could you enter to stage the latest changes?

```
$ git add src/io.cpp
```

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

b) What command(s) could you enter to unstage the changes *after* running the previous command(s), i.e (a)?

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

b) What command(s) could you enter to unstage the changes *after* running the previous command(s), i.e (a)?

```
$ git reset HEAD src/io.cpp
```

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

c) What command(s) could you enter to commit the latest changes *after* previous unstaging command(s), i.e. (b)?

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

c) What command(s) could you enter to commit the latest changes *after* previous unstaging command(s), i.e. (b)?

```
$ git commit -a -m "latest changes"
```

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

d) What command(s) could you enter to discard the changes in the working directory *before* doing the above commit, i.e. (c)?

Question 2

Suppose running the **'git status'** command produced the following output:

```
# On branch master
# Changes not staged for commit:
#
#       modified:   src/io.cpp
```

d) What command(s) could you enter to discard the changes in the working directory *before* doing the above commit, i.e. (c)?

```
$ git checkout -- src/io.cpp
```

Question 3

a) What is the output of the following Python 3 program?

```
num1 = 5
if num1 >= 91:
    num2 = 3
else:
    if num1 < 6:
        num2 = 4
    else:
        num2 = 2
x = num2 * num1 + 1
print (x,x%7)
```

Question 3

a) What is the output of the following Python 3 program?

21 0

```
num1 = 5
if num1 >= 91:
    num2 = 3
else:
    if num1 < 6:
        num2 = 4
    else:
        num2 = 2
x = num2 * num1 + 1
print (x,x%7)
```

Question 3

a) What is the output of the following Python 3 program?

```
list2 = ['B','C','A']
list2.reverse()
list2.append('S')
list2.sort()
list2.reverse()
print(list2)
```

Question 3

a) What is the output of the following Python 3 program?

```
list2 = ['B','C','A']
list2.reverse()
list2.append('S')
list2.sort()
list2.reverse()
print(list2)
```

```
['S', 'C', 'B', 'A']
```

Midterm Exam – Part II

Is set as your Assignment 2

Python

Using a Debugger

IDE – Integrated Development Environment

- Microsoft Visual Studio
  - Available for UOIT students from DreamSpark
    - https://uoitsci.onthehub.com
  - 2015 includes Python Tools for Visual Studio

IDE – Integrated Development Environment

- Wing IDE
  - Installation on Ubuntu:

```
$ sudo dpkg -i wingide-101-5-5.1.8-1.x86_64.deb
$ sudo apt-get install -f
```