BSc (Hons) in Information Technology
IT– Year 2

Lab Exercise 5

IT2060 – Operating Systems and Systems Administration

Environment Variables

1) Write down the output for following commands: (Remember Unix is *case sensitive*, therefore type them as they are.)

```
echo $PATH    ……………………………………………………………………
echo $HOME    ……………………………………………………………………
echo $USER    ……………………………………………………………………
echo $PWD     ……………………………………………………………………
echo $PS1     ……………………………………………………………………
```

a) Did you assign the values that you noted down for the above variables? Yes/No No

b) If not, who assigned those values for the above variables? ……………………………

Creating / Accessing Simple Variables

2) Assign and print following variables

```
labclass="This is OSSA Lab Class"
echo $labclass
num=`expr 13 + 5`
echo $num
num=`expr 13 \* 5`          ←note the backslash before the multiplication sign
echo $num       ⍰
num1=`expr $num \/ 7`       ←note the backslash before the division sign
echo num1       ⍰
num1=`expr $num \% 7`       ←note the backslash before the remainder sign
echo num1       ⍰
num1=`expr $num1 + 8`       ←note the variable name
```

3) Create a shell script **usrinfo.** Define variables call “name”, “address”, “country” inside the **usrinfo.** Assigned your name, address, and country respectively in to the variables. Write statements to print contents of the variables on the screen. Save and quit the script.

4) Give full permission to the script you created in part 3) i.e. **usrinfo.** Change permission of the same file, so that **others** have no access to the file. (Hint: You can use **chmod** command to enable and disable file permissions)

5) Write down the commands that can be used to execute a shell script,

a) in which user doesn’t have execution permission ……………………………………………

b) in which user has execution permission ……………………………………………

Setting *PATH* to search for newly created executables

6) Can you execute the script by just typing its name (like other Linux commands).Yes/No No
If 'No,' then set the *PATH* variable to search your present working directory (**pwd**) for executable files. Follow the command sequence given below to set the *PATH*.

```
echo $PATH                                    ←    check the existing path variable
pwd                                           ←    to get your current working
directory
PATH=$PATH:<presentworkingdirectory>          ←append your current working directory
echo $PATH                                    ←check the modified PATH variable
```

Execute the script by just typing its name.

Exporting Variables

7) Create a variable call "*amount1*" in the shell prompt and assign value **100**. Follow the command sequence given below

```
bash
echo $amount1
```

Can you get the value of "*amount1*"?   Yes/No

Type **ps –Hl** to identify relationship between subshells. (Please repeat this **ps –Hl** after each question to see this relation ship.
Now exit the sub-shell by typing "**exit**".

8) Create a variable call "*amount2*" in the shell prompt and assign value **200**. Follow the command sequence given below

```
export amount2
bash
echo $amount2
```

Can you get the value of "*amount2*"?   Yes/No

Run **bash** again to enter into a new subshell. Can you access the *$amount* variable from this shell? Run **ps –Hl**. Now exit the sub-shells by typing "**exit**". Can you access the *$amount* from parent shell?

Shell Programming

9) Make a directory **scripts** inside your working directory. Change your directory to new directory (by typing **cd scripts**). Create a file ***script1*** and enter the following lines to the file ***script1***.

```
#!/bin/bash

echo Welcome to
SLIIT Server echo
Today is :
date
```

Save the file and quit the editor.

10) Type **sh *script1*** and observe the output.

11) Why can’t you type **script1** to run it? (Hint: Type ls –l script1 and check the file permissions.)
…………………………………………………………………

12) Run **chmod u+x script1** to enable the rights for execution. Now run the command by typing its name **script1**. Can you run it? Yes/No

13) If you can’t run your script in previous step, type the *absolute path* of your

script to run it. Do you need to type *absolute path* to run normal UNIX commands?
Describe the reason: ………………………………………………………………………………………

3. nano usrinf.sh

to save and exit - Press ctrl + x
press y for yes
Enter