



INSTITIÚID TEICNEOLAÍOCHTA, SLIGEACH
INSTITUTE OF TECHNOLOGY, SLIGO

School of Engineering and Design

Head of School: Mr Shane Fanning

Exam Series Semester 2

Academic Year 2015/2016

Module Title:

Module Code:

Shell Scripting

COMP06212

Programme Code(s):	Programme(s) Name(s)	Year(s)	FT /PT
SG_KCOMP_H08	BSc. (Hons) In Computing	1	FT

Internal

Examiner(s): Dana Vasiloaica

External

Examiners: Damien Costello

Instructions to Candidates

Time Allowed: **3 Hours**

Number of Questions on Paper: **3 Questions in Part A**
6 Questions in Part B

Number of Questions to be attempted: **Any 2 Questions from Part A**
Any 4 Questions from Part B

Compulsory Question: **None**

Any Other Special Instructions:

Please answer any 2 questions from Part A and any 4 questions from Part B

To submit your work, please upload to Moodle each and every script that you created as part of this exam – go to the Moodle page and follow the link reading UPLOAD YOUR EXAM HERE (this link is located at the top of the Shell Scripting Moodle page, in section labeled SUMMER EXAM).

Remember that it is your responsibility to submit your work! Please double check to ensure that the files are uploaded in Moodle. Do not leave the exam room before you ensure that your work is uploaded! No uploads means no marks! No exceptions!

Final note: please ~~save your work~~ as you go along. It is your responsibility to ensure that you do not accidentally delete or damage the scripts you are working on. You will only receive marks for the scripts you upload in Moodle and not for the work that you somehow lost during the exam!

Part A [60 marks]

Please answer any 2 of the following questions:

Part A - Question 1 [30 marks]

Create a shell script named UsersScript.sh
The script should perform the following tasks:

1. Check if the number of command line arguments equals 2.
 - a. If it is, create an array with a name of your choice. Place the first command line argument in the first element of the array. Place the second command line argument in the second element of the array.
 - b. Else write an error message to the terminal and terminate the script.
2. Create a function named `create()`.

This function should check if the first argument passed to the function is the exact same as the second argument passed to the function.

If the 2 arguments are different, then:

 - a. Create a group with the name specified by the first argument.
 - b. Create a user with the username specified by the second argument, and place this user in the group you have just created above.
 - c. Then write a message (of your choice) to the terminal saying that both the group and the user were successfully created.

- d. Finally, delete both the group and the user (from the system).

If the 2 arguments supplied to the function are identical, then simply display an error message (of your choice) to the terminal.

3. Create a function named **check()**.

This function should:

- a. Ask the user to input a user name.
- b. Capture that username and check if a user with that username exists.
- c. If the username exists, then write a message of your choice to the terminal.

Example: User is already in the system

- d. If the user does not exist, then write a different message (again your choice) to the terminal.

Example: User does not exist

4. Use a select statement to display the following options to the user:

1. Create and Delete
2. Check if user exists
3. Quit

- a. If the user chooses option 1 (Create and Delete), then call the function named **create()** and pass the first and second elements of the array as its arguments (the first element of the array is the first argument of the function; the second element of the array is the second argument of the function).
- b. If the user chooses option 2 (Check if user exists), then call the function named **check()**.
- c. If the user chooses option 3 (Quit), then simply exit the application.
- d. In all other cases, display an error message of your choice.

To test your script, issue the following command (at the terminal):

```
sudo bash UsersScript.sh mygroup myuser
```

[30 marks;

2 marks for creating the array,
10 marks for the **create()** function,
8 marks for the **check()** function,
5 marks for select,

5 marks for creating a script which runs as intended and with no errors]

Part A - Question 2 [30 marks]

Create a text file named *testFile*. Place some lines of text in there (your choice how many), but please write the word *hi* somewhere in the file.

Create a shell script and name it *Math.sh*

Your tasks are as follows:

1. Create a function named **square()**.
 - a. Check if the number of arguments supplied to the function is 1.
 - b. If it is, then the function should calculate and display (on the terminal) the value of the argument squared (multiply the value of the argument supplied to the function by itself).
 - c. If the number of arguments supplied to the function is not 1, then the function should display an error message of your choice.
2. Create a function named **divide()**.
 - a. Create a read-only variable named *roVar*. Set the value of this read-only variable to 10.
 - b. Find out how many lines of text contain the word *hi* within the text file named *testFile*. Save this number (i.e. the number of lines you counted above) into a variable named *countVar*.
 - c. Finally, divide *countVar* by *roVar* and display the result at the terminal.
3. Use a **select** statement to display the following options to the user:
 1. Square
 2. Divide
 3. Quit
 - a. If the user chooses option 1 (Square), then call the function named **square()** and supply a number of your choice as argument.
 - b. If the user chooses option 2 (Divide), then call the function named **divide()**.
 - c. If the user chooses option 3 (Quit), then simply exit the application.
 - d. Display an error message for all other options.

[30 marks;

10 marks for the **square()** function,

10 marks for the **divide()** function;

5 marks for the **select**;

5 marks for creating a script which runs as intended and with no errors]

Part A - Question 3 [30 marks]

Create a text file named *file1* and place some text into it.

Create a second text file named *file2* and place some text into it.

Create a shell script and name it *FileInfo.sh*

The script should perform the following tasks:

Use a select statement to display the following options to the user:

1. List some directories
2. Check some files
3. Quit

1. If the user chooses option 1 (List some directories), then you should use a for loop to display at the terminal the names of all directories (within the current directory) whose names start with the letter specified as the script's command line argument.
2. If the user chooses option 2 (Check some files), then your script should:
 - a. Ask the user to input a file name.
 - b. Capture the user input and save it in a variable named with a name of your choice.
 - c. Ask the user to input another file name.
 - d. Capture the user input and save it in a variable named with a name of your choice.
 - e. Check if the first file is older than the second one and display the result at the terminal (use a message of your choice).

Example: file1 is older than file2
3. If the user chooses option 3 (Quit), then simply exit the application.
4. Display an error message for all other options.

To test your script, issue the following command (at the terminal):

```
bash FileInfo.sh D
```

When you are asked to input a file name, type *file1* and *file2*

[30 marks;

5 marks for the select statement,

10 marks for code behind *List some directories*,

10 marks for the code behind *Check some files*,

5 marks for creating a script which runs as intended and with no errors]

Part B [40 marks]

Please answer any 4 of the following questions:

Part B - Question 1 [10 marks]

Create a shell script named Script1.sh

The script should display on the terminal the following information:

1. Count how many processes are running on your system at the moment. Please ensure that you count them all (and not just your processes).
2. List all the processes running, while showing details about each one.
3. Count how many users are logged in at the moment.
4. List all the users logged in at the moment.
5. Display today's date.

[10 marks; 2 marks each]

Part B - Question 2 [10 marks]

Create a shell script named Script2.sh.

The script should perform the following actions:

1. Ask the user to input a number.
2. Save that number in a variable of your choice.
3. Ask the user to input another number.
4. Save this second number in a variable of your choice.
5. Compare the first number with the second and display (at the terminal) if the first number is greater, less than, or equal to the second number.

[10 marks;

1 mark each for points 1, 2, 3 and 4,
6 marks for point 5]

Part B - Question 3 [10 marks]

Create a shell script named `Script3.sh`

The script should display on the terminal the following information:

1. The current operating system type.
2. The name of the default directory (home directory).
3. The shell name.
4. The number of columns on your screen.
5. Your path settings.

[10 marks; 2 marks each]

Part B - Question 4 [10 marks]

Create a text file named *myfile* and place the following text into it:

Hello, my name is and my student ID is

Create a shell script named `Script4.sh`

The script should perform the following actions:

1. Display at the terminal the lines of text containing the word *student* within the file named *myfile*.
2. Create a hard link to the file *myfile*. Name it with a name of your choice.
3. Create a symbolic link to the file *myfile*. Name it with a name of your choice.
4. Copy the file *myfile* to a file named *mycopy*.
5. Rename the file *myfile* to *mynewfile*.

[10 marks; 2 marks each]

Part B - Question 5 [10 marks]

Create a file named *somefile* and place some text in there (your choice what text)

Create a shell script and name it *Script5.sh*

The script should perform the following actions:

1. Change the permissions of *somefile* according to the following requirements:
 - a. Give the owner write and read permissions.
 - b. Give everyone execute and read permissions.
 - c. Give the group execute permissions only.
2. Create a directory named *mydir*.
3. Change directory into *mydir*.
4. Next, change directory back to where you started from.
5. Delete the directory *mydir*.

[10 marks; 2 marks each]

Part B - Question 6 [10 marks]

Create a shell script named *Script6.sh*

The script should perform the following tasks:

1. Create an array named with a name of your choice.
2. Ask the user to input 3 numbers.
3. Capture (one by one) each number and place them at consecutive indexes within the array – I suggest you start at index 0.
4. Use a loop of your choice to display the numbers stored in the array.

[10 marks;

1 mark for creating the array,

3 marks for capturing the 3 numbers and store them in the array,

6 marks for the loop]