EECS 2031 - Lab Test 1 – makeup

**Description**

This lab test is mainly based on Lab 3 and 4 and considering the knowledge of Lab 1 and 2. In particular, contrary to Lab 3 and 4, the shell script only processes a command which is passed as an argument to the shell script. **Make sure you have read all Lab 3 and 4 requirements.**

What to do

* **DO NOT USE LAB4 SOLUTION AS YOUR START CODE. IF YOU USE THE LAB 3 OR 4 SOLUTION, THE SOFTWARE USED FOR ACADEMIC INTEGRITY WILL REJECT YOUR SUBMISSSION.**
* Create a new file name labtest1\_makeup.sh using your favorite editor so that it implements the functionality described in the Requirements section below.
* Test your shell script correctly such that it produces the required outputs based on the defined requirements.
* Submit your solution electronically using the command

submit 2031 labtest1\_makeup labtest1\_makeup.sh

* **The due date is at 6:00PM without any exception and extension**.
* During the lab test, you do not be online in Zoom.
* I do not answer any question during the lab test.
* You may submit your solution more than once. Additional documentation about the submit command can be viewed by typing man submit.

General Requirements

* Your script should be run under Born Shell.
* Enter your first name + last name + student number as a comment at the top of the script.
* Enter comments for all parts of your script to explain the functionality of the code.
* Only use covered commands in lectures. You may need to check the “man” page of covered commands.
* Remove any temporary file you may have generated before exit.
* **Make sure the output of all commands are exactly the same as the expected output. Consider spaces, character lowercase/upper cases to be the same as the expected output**.
* In case of normal exit return 0, otherwise return 1.

Rec Files

You can find all rec files you need for this lab tests from this link using the “wget” command.

<https://www.eecs.yorku.ca/~nsajadi/Teaching/EECS2031/labtest1/recfiles.zip>

**Note1:** that we may use a different file set for grading but the format is the same as the above link.

**Note** **2**: Each student received different marks for different activities in each course. To get the total mark for each student for each course you need to sum the list of numbers.

Specific Requirements

We want to implement 4 commands. Here is the list of new commands:

##########

# Wrong parameter cases

# **Marking: Exact output: 1 mark**

##########

labtest1.sh

**You should enter the path name for course files and at least one command.**

**Use: labtest1.sh path command [arg1 arg2 ...]**

**For the list of all commands use:**

**Example1: labtest1.sh . h**

**For the list of number of registered students in each course use:**

**Example2: labtest1.sh . creg**

labtest1.sh ~

**You should enter the path name for course files and at least one command.**

**Use: labtest1.sh path command [arg1 arg2 ...]**

**For the list of all commands use:**

**Example1: labtest1.sh . h**

**For the list of number of registered students in each course use:**

**Example2: labtest1.sh . creg**

labtest1.sh ~/tmp cl

**There is no readable \*.rec file in the specified path or its subdirectories.**

##########

# h: prints the help message.

# **Marking: Exact output: 1 mark**

##########

labtest1.sh ~ h

**Here are defined commands:**

**creg: give the list of course names with the total number of students registered in each course.**

**stc ######: gives the name of all course names in which the student with ###### id registered in.**

**gpa ######: gives the GPA of the student defined with id ###### using the following formula: (course\_1\*credit\_1 + . . . + course\_n\*credit\_n) / (credit\_1+ . . . + credit\_n)**

**cstats: Gives the course/student statistics.**

**h: prints the current message.**

##########

# **creg**: give the list of course names with the total number of students registered in each

# course.

# **Hint**: Loop over each sound course files and look for patterns of course names and

# credits.

# **Marking: Exact output: 3 marks**

##########

labtest1.sh ~ creg

**1. In "SIGNALS AND SYSTEMS", 4 students register. The course has 4 credits.**

**2. In "OPERATING SYSTEMS", 3 students register. The course has 3 credits.**

**3. In "JAVA PROGRAMMING", 4 students register. The course has 4 credits.**

##########

# **stc ######**: gives the name of all course names in which the student with ###### id

# registered in.

# **Hint**: Check the student id length first, then the same as the “creg” command loop

# through all found courses and look for existence of the student with id ######.

# **Marking: Exact output: 3 marks**

##########

labtest1.sh ~ stc

**The student id should be 6 numbers.**

labtest1.sh ~ stc 123456

**The student with id: 123456, is registered in the following courses:**

**1. JAVA PROGRAMMING which has 4 credits.**

**2. SIGNALS AND SYSTEMS which has 4 credits.**

**3. OPERATING SYSTEMS which has 3 credits.**

labtest1.sh ~ stc 346519

**The student with id: 346519, is registered in the following courses:**

**1. JAVA PROGRAMMING which has 4 credits.**

labtest1.sh ~ stc 01245

**The student id should be 6 numbers.**

labtest1.sh ~ stc 012450

**The student with id: 012450 is not registered in any course.**

labtest1.sh ~ stc 0124500

**The student id should be 6 numbers.**

##########

# **gpa** ######: gives the GPA of the student defined with id ###### using the following

# formula: (course\_1\_mark\*credit\_1 + … + course\_n\_mark\*credit\_n)/ (credit\_1+ … + credit\_n)

# **Hint**: First you need to find the list of all files the given student is registered in using the

# “grep” command then find the student in each file. Loop over all received marks at front

# of the student id to find the course\_?\_mark. Keep track of all credits then you can use the

# given the formula to calculate the GPA for the given student.

# **Marking: Exact output: 4 marks**

##########

labtest1.sh ~ gpa 123456

**The GPA for the student with id: 123456 is 90.**

labtest1.sh ~ gpa 346519

**The GPA for the student with id: 346519 is 59.**

labtest1.sh ~ gpa 589999

**The GPA for the student with id: 589999 is 75.**

labtest1.sh ~ gpa 243567

**The GPA for the student with id: 243567 is 54.**

labtest1.sh ~ gpa 6519

**The student id should be 6 numbers.**

labtest1.sh ~ gpa 01245

**The student id should be 6 numbers.**

labtest1.sh ~ gpa 012450

**The student with id: 012450 is not registered in any course.**

labtest1.sh ~ gpa 0124500

**The student id should be 6 numbers.**

##########

# **cstat**: gives the GPA of the student defined with id ###### using the following

# formula: (course\_1\_mark\*credit\_1 + … + course\_n\_mark\*credit\_n)/ (credit\_1+ … + credit\_n)

# **Hint**: First find the name of all course names and make the header of the desired output

# given bellow. The find the unique list of students and iterate over the unique list

# of students. The rest of the code is almost the same as the “gpa” command with proper formatting.

# **Marking: Exact output: 8 marks**

##########

labtest1.sh ~ cstat

