# CSCI 3308 Software Development Methods and Tools, Fall 2017

**Homework # 1**

## Topic: Bash Shell Scripts & Regex

**Due Date:** 5pm on September 20, 2017

**Submission:** Submit via the Link embedded in Week Three materials in Moodle

## Objectives

- Write a bash shell script with bash shell commands to loop through a data file
- Write a bash shell script using UNIX commands like "awk"
- Practice using Regex commands to parse text
- Gain more experience in pair-programming collaboration [optional]

## Part 1 Bash Shell Scripts

Submit two separate bash shell scripts that

1. Reads the following data file,
2. Calculates the average of the scores for each record
3. Sorts the output by last name then first name
4. Formats the output as shown below.

The objective of writing two scripts is to see that there are multiple correct solutions to such problems. One solution should use the awk tool, and the other should use bash commands (bash scripting).

Write a program that reads in a data file (get filename from command line arguments) and prints out the average for each person, as shown in the "report" below based on the example data file below.

***Note: we will accept both rounded and truncated averages.***

**Report:**

123456789 Lee Johnson 72 85 90
999999999 Jaime Smith 90 92 91
888111818 JC Forney 100 81 97
290010111 Terry Lee 100 99 100
199144454 Tracey Camp 77 84 84
299226663 Laney Camp 70 74 71
434401929 Skyler Camp 78 81 82
928441032 Jess Forester 85 80 82
928441032 Chris Forester 97 94 89

An example data file is shown below:

71 [299226663] Camp, Laney
80 [434401929] Camp, Skyler
81 [199144454] Camp, Tracey
93 [928441032] Forester, Chris
82 [928441032] Forester, Jess
92 [888111818] Forney, JC
82 [123456789] Johnson, Lee
99 [290010111] Lee, Terry
91 [999999999] Smith, Jaime

## Part 2 Regex

Download the Regex Practice Data from the Moodle site.

For each of the questions listed below, submit the regex expression necessary to calculate the answer. Please refer to the requirements section in this document for details on submission expectations.

The command grep and egrep are your friends (hint: egrep treats { } differently than grep).

Be sure to check for word boundaries in your answers '\b' where appropriate.

*Hint: Pipe answers to "wc –l" to get the count.*

There are multiple correct solutions for each of the ones below. I encourage you to explore the variant answers as well for a better understanding of regex.

1. How many lines end with a number?
2. How many lines do not start with a vowel?
3. How many 12 letter (alphabet only) lines?
4. How many phone numbers are in the dataset (format: '_ _ _-_ _ _-_ _ _ _')?
5. How many city of Boulder phone numbers (e.g. starting with 303-_ _ _-_ _ _ _)?
6. How many begin with a vowel and end with a number?
7. How many email addresses are from geocities? (e.g. end with'geocities.com')?
8. How many records have incorrect email addresses (lines with an @ in it but formatted incorrectly)? An email address has a user-id and domain names can consist of letters, numbers, periods, and dashes. An email address should have a top-level-domain (something.top-level-domain). Top-level-domains are of the form: *com, org, edu* etc.,

### Requirements

1. Scripts must be bash files named
   a. ● Grades.sh
   b. ● GradesAwk.sh
   c. ● RegexAnswers.sh

# CSCI 3308 Software Development Methods and Tools, Fall 2017

**Homework # 1**

2. At the top of your scripts, include comment with your name (and your partner's name if you pair program).

3. For all scripts, read in the name of the data file from command-line arguments. (The file names should not be hard coded in the scripts.) We will test all the three scripts with additional data files that have different names

***4.*** Grades scripts:

● The data files for the grades scripts will be the same format as shown, though it may have more or less lines in the file. All students have 3 grades in the data files.

● Print out the data with the average, the ID in square brackets, then the last name, comma, space, first name.

● The output also needs to be sorted, first based on the last name. If the last name is the same, sort then on the first name. If the person has the same last name and first name, then sort based on the ID. All IDs are unique in the file.

● If the program is run without one filename as the command-line argument, print out the usage statement:

Usage: Grades.sh filename
Or
Usage: GradesAwk.sh filename

5. Regex programs

● Each line of output should map to the question. There are seven questions so you should only have 7 lines of output which is the output from calling 'wc –l' If you do not know how to do one of the answers print out this placement so that the rest of your answers align in the output:

echo "0"

● If the program is run without one filename as the command-line argument, print out the usage statement:

Usage: RegexAnswers.sh filename

6. You will only submit a single zipped file containing all three of your script files. If you are pair-programming only ***one*** of you will submit.

If you are working alone, name the zip file using the following template:
Lastname_HW1.zip

If you are pair programming, then name the file using this template:
Lastname1_Lastname2_HW1.zip

# CSCI 3308 Software Development Methods and Tools, Fall 2017

## Homework # 1

**Note: All the 3 scripts should be runnable from command line where filename is given as an argument. If a script doesn't execute or doesn't provide the right output, then points will be deducted.**