# ECE 364 Prelab Assignment 08
Bash Shell Scripting

**Passing this lab will satisfy course objectives CO1**

## Instructions

- You must meet all *base requirements* in the syllabus to receive any credit.
- Work in your Prelab08 directory, and copy all files from ∼ee364/DataFolder/Prelab08 into your working directory:

  ```
  cp -r ∼ee364/DataFolder/Prelab08/* ./
  ```
- Remember to add and commit all **required** files to SVN. **We will grade the version of the file that is in SVN!**
- Do *not* add any file that is *not* required. **You will lose points if your repository contains more files than required!**
- Make sure you file compiles. **You will not receive any credit if your file does not compile.**
- Name and spell the files exactly as instructed. Your scripts will be graded by an automated process. **You will lose some points, per our discretion, for any file that does not match the expected name.**
- Make sure your output from all functions match the given examples. **You will not receive points for a question whose output mismatches the expected result.**

# Bash Shell Scripting

**Required Files** `getCircuitsByProject.sh`, `getStudentsByProject.sh`, `getCircuitsByStudent.sh`, `getProjectsByStudent.sh`, `getComponentUseCount.sh`, `getMoreUsedComponent.sh`

## Description

Create the required Bash files and do all of your work in them. These are the only files you need to submit. Refer to the **Bash File Structure** section below for a reminder on the requirements.

Students in some university are collaborating on several projects, where in each projects they are building multiple circuits using four component types: Resistors, Inductors, Capacitors and Transistors. You are given some files in a sub-folder, called `maps`, indicating that these files contain mapping information between two collections. These files are:

- `students.dat` containing student names and their IDs. Note that, throughout this assignment, the format of the student name should match what is used in this file.
- `projects.dat` containing project IDs and the circuits IDs that belong to each project.

In addition, you are given another sub-folder called `circuits` containing multiple files, where each file holds information about a specific circuit that may have been used in one or more projects. The format of each file name in the folder is `circuit_<circuitID>.dat` and it contains two types of information:

- The students IDs that have collaborated in building that circuit.
- The unique set of component IDs used in that circuit.

While the format of each circuit file is consistent, the number of students participating in building each circuit varies, and so does the number of components used in that circuit. Moreover, some circuits may not have uses all component types, i.e. they may have not used transistors, or inductors, etc. Examine all of the files before you begin, to make sure you are comfortable with their format.

Note: You may hardcode the names of the two sub-folders along with the names of the files in the `maps` sub-folder. However, you cannot hardcode any of their content, nor can you hardcode the circuit IDs.

## Requirements

1. [**10 pts**] Create a Bash file named `getCircuitsByProject.sh`, that takes in a project ID, and prints out a *sorted* list of all Circuit IDs used in that project, with *no duplicate entries.* The Project ID will be passed as a command line argument, e.g.:

```
# The following ID produces 20 circuits.
bash getCircuitsByProject.sh 082D6241-40EE-432E-A635-65EA8AA374B6
13-2-03
15-5-65
16-7-59
...
74-1-27
81-5-42
81-9-47
```

2. [**20 pts**] Create a Bash file named `getStudentsByProject.sh`, that takes in a project ID, and prints out a *sorted* list of the names of all the students who have participated in that project, with *no duplicate entries.* The Project ID will be passed as a command line argument, e.g.:

```
# The following ID produces 50 names.
bash getStudentsByProject.sh 082D6241-40EE-432E-A635-65EA8AA374B6
Alexander, Carlos
Allen, Amanda
Bailey, Catherine
...
Wood, Kevin
Wright, Eric
Young, Frank
```

3. [**15 pts**] Create a Bash file named `getCircuitsByStudent.sh`, that takes in a student's name, and prints out a *sorted* list of all Circuit IDs that the student has participated in, with *no duplicate entries.* The name will be passed as a command line argument, e.g.:

```
bash getCircuitsByStudent.sh "Scott, Michael"
20-5-40
21-1-59
30-6-36
53-9-58
56-9-86
72-2-77
79-3-65
79-7-09
```

4. [**15 pts**] Create a Bash file named `getProjectsByStudent.sh`, that takes in a student's name, and prints out a *sorted* list of all Project IDs that the student has participated in, with *no duplicate entries.* The name will be passed as a command line argument, e.g.:

```
# The following name produces 18 IDs.
bash getProjectsByStudent.sh "Scott, Michael"
075A54E6-530B-4533-A2E4-A15226BE588C
08EDAB1A-743D-4B62-9446-2F1C5824A756
0F1FABFA-E112-4A66-A0B0-B7A2C14AD39A
...
D7EFB850-9A34-41B0-BD9D-FBCDF4C3C371
D88C2930-9DA4-431F-8CDB-99A2AA2C7A05
DE06228A-0544-4543-9055-A39D19DEDFA4
```

5. [**10 pts**] Create a Bash file named `getComponentUseCount.sh`, that takes in a component ID, and prints out the number of circuits that use it. The component ID will be passed as a command line argument, e.g.:

```
bash getComponentUseCount.sh CPF-254
6
```

6. [**10 pts**] Create a Bash file named `getMoreUsedComponent.sh`, that takes in two component IDs, and prints out the ID of the component that is used in more circuits. The component IDs will be passed as a command line argument, e.g.:

```
bash getMoreUsedComponent.sh CPF-254 VGT-864
CPF-254
```

## Final Check!

You are given a file that checks for the exact spelling of the required files. Make sure you run this file before your final commit. Remember that **you will lose some points for any function that does not match the expected name.**

## Bash File Structure

The following is the expected file structure that you need to conform to:

```
#######################################################
#   Author:     <Your Full Name>
#   email:      <Your Email>
#   ID:         <Your course ID, e.g. ee364j20>
#   Date:       <Start Date>
#######################################################

# Write your sequence of statements here.
...
...

exit 0
```