**Script 1**

Take first argument handed in with the script and store it in the variable ‘INPUT’

Set variable ‘OLDIFS’ to the current system IFS (IFS is the internal field separator that’s used for word splitting)

Set the system IFS to ‘, ‘

Check if the argument passed in is a file, if not exit the script with error code 99 and print out to the user that the file was not found.

While loop that will run until the last line of the ‘INPUT’ file and store 4 variables which are ‘login’, ‘idNum’, ‘course’ and ‘fullName’. Therese variable derive from each line from the ‘INPUT’ file with each variable being separated by ‘,’ (the new IFS) in each line.

Find the user from the /etc/passwd directory (this directory store all the users) that matches the ‘login’ variable.

If found, comment will be added to the user using the ‘usermod’ command. The comment will be made up of variable ‘full name’ and ‘course’

Else if the user does not exist, then add the user with the comment deriving from the variable ‘fullname’ and ‘course’. Making the home directory of the user ‘/home/<variable login>’. The -m ensure that the home directory if doesn’t exist, will be made. -U means create a user group with the same name as the user. The shell of the user will be /bin/bash and the username are the variable ‘login’.

Then set the password for this user to be its id number by echoing it in this format ‘username:password’ to be used in the ‘chpasswd’ command.

Then force the password to change when user initially log in.

The command ‘edquota’ will set the disk quota per user base on the quota allocated to the user darrelll

The use case for this is to create new users based on the file inputted into the bash script. The user will then have its system quota set. If user already exist, it will only add comment to the user. The input file structure will have to be formatted like this per line (username, id number, course enrolled, fullname)

**Script 2**

Take first argument handed in with the script and store it in the variable ‘INPUT’

Set variable ‘OLDIFS’ to the current system IFS (IFS is the internal field separator that’s used for word splitting)

Set the system IFS to ‘, ‘

Check if the argument passed in is a file, if not exit the script with error code 99 and print out to the user that the file was not found.

While loop that will run until the last line of the ‘INPUT’ file and store 4 variables which are ‘login’, ‘idNum’, ‘course’ and ‘fullName’. Therese variable derive from each line from the ‘INPUT’ file with each variable being separated by ‘,’ (the new IFS) in each line.

Find the user from the /etc/passwd directory (this directory store all the users) that matches the ‘login’ variable.

If found, comment will be added to the user using the ‘usermod’ command. The comment will be made up of variable ‘full name’ and ‘course’

Else if the user does not exist, then add the user with the comment deriving from the variable ‘fullname’ and ‘course’. Making the home directory of the user ‘/home/<variable login>’. The -m ensure that the home directory if doesn’t exist, will be made. -U means create a user group with the same name as the user. The shell of the user will be /bin/bash and the username are the variable ‘login’.

Then set the password for this user to be its id number by echoing it in this format ‘username:password’ to be used in the ‘chpasswd’ command.

Then force the password to change when user initially log in.

Then append another group to the user. The group added is called ‘docker’. The -a is needed otherwise all existing group will be removed from the user.

The use case for this is to create new users based on the file inputted into the bash script. The user will then be added to the ‘docker’ group. If user already exist, it will only add comment to the user. The input file structure will have to be formatted like this per line (username, id number, course enrolled, fullname)

**Script 3**

Take first argument handed in with the script and store it in the variable ‘INPUT’

Set variable ‘OLDIFS’ to the current system IFS (IFS is the internal field separator that’s used for word splitting)

Set the system IFS to ‘, ‘

Check if the argument passed in is a file, if not exit the script with error code 99 and print out to the user that the file was not found.

While loop that will run until the last line of the ‘INPUT’ file and store 1 variables which is ‘login’.

Find the user from the /etc/passwd directory (this directory store all the users) that matches the ‘login’ variable.

If found, the user’s password will be locked. Password are locked by changing it to a value to a value that matches no possible encrypted value (it adds a ´!´ at the beginning of the password).

The use case for this is to disable access of a list of user but not getting rid of the user completely or disabling it.