**PROJ 8 NOTES**

* **1 - Archiving User Data from Directory Path**

The function begins by taking in our input as an argument -

**my ($user\_directory) = @\_;**

Then we create a new zip file to store the user directory data -

**my $member = $zip->addTree($user\_directory, $user\_directory);**

Our member variable is assigned to our zip variable, which called the addTree method. This method will take the contents of one directory (first argument (user\_directory)) and determine the organization structure of this archive, which can be based off another argument (user\_directory).

**unless ($zip->writeToFileNamed("$user\_directory.zip") == AZ\_OK)**

**{ die "Error creating $user\_directory archive - " . $zip->errorString; }**

This section is an error prevention check. AZ\_OK is a built\_in constant included in Archive::Zip, used to ensure the specified action is performed correctly. The action is that we are writing our zip variable to a new zip file, which will go to the same directory location as the original. If this action does not perform correctly, an error message pops up.

After all has been calculated and archived, we can put our directory name from user\_directory into our deleted\_users array, for storing usernames -

**push @deleted\_users, $user\_directory;**

* **2 - Deleted User Report**

To start off, in order to keep track of all of our usernames, we can collect them into an array, created towards the top of the program -

**my @deleted\_users;**

Next we create a variable to cycle through each entry in our array with a foreach loop -

**foreach my $user (@deleted\_users) {**

**my ($username) = $user =~ m|/([^/]+)$|;**

**my $zip\_locate = "$username.zip";**

**print "Username - $username\n";**

**print "ZIP Location - $zip\_locate\n\n";**

Basically, for each going over of the loop, we assign $username to our current loop through ($user), and assign that to a regular expression. Given that the array is full of file path names, and we only want to display the username, we can use this regular expression to specify the username section of the directory path.

We can then print out our username and ZIP location. Afterward, it brings us back to the main function and asks for confirmation if we want to leave. If yes (y), the program will end. If no (n), The while loop will bring us back to the home menu.

* **3 - Prompting Directory Pathway**

The program takes in an entry from the user for a directory pathway to the user's directory folder, starting from /home. this is what the if statement is clarifying -

**if ($user\_directory =~ /^\/home\/\w+$/)**

The square bracketed section specifies which characters can be used, in this case -

**/home … / … w+ for an additional word (the username)**

Under these conditions, the archive and delete functions should run for the user, given the user input as arguments to be processed.