# Analysis

## Scenario

The scenario describes a quiz that tests student’s knowledge on several topics. It asks that students must register an account before taking the quiz, which will be saved into an external file. To register, they must enter their name, age and year group.

I need to create a program asking the user to enter their name, age and year group.

It **also** says to make the username from the first three letters of their name and their age.

I just need to create an anagram of their name and age, so for example, mine would be ‘Ars16’. And I assume that the user must also create their own password.

I have to store their username, password, year group, topic they chose, their score, and difficulty they chose.

I’ll need multiple files. Or, I could store it all into one CSV file. Even so, I’ll need TWO programs, one to register, and one to login and actually take the test.

I have to load the questions and answers in an external file. It ALSO says the program must show the user’s score, percentage and grade.

So I WILL need multiple files.

A GUI wasn’t specified, so I’ll use good old CLI.

And I’m not using TkInter.

Not here, anyway.

# Plan

The files I’ll need are: registration program, quiz program, external questions file(s), external user file

# Design

## Success Criteria

* Create a program to register users and store their info in a CSV file
* Create a program to be the actual quiz and use the data from the aforementioned CSV file

## I’m thinking it should go like this for the registration…

Enter your name:

Enter your age:

Enter year group:

Enter a password:

Welcome, (name).

Your username is: (first three letters of name followed by age)

Your password is: (password)

## …and this for the actual quiz.

Enter your username:

Enter your password:

Question 1: (question name)

etc.

# Pseudocode

## Registration

Ask "Enter your name: " and wait for user input

Store to variable name

Ask "Enter your age: " and wait for user input

Store to variable age

Ask "Enter your year group: " and wait for user input

Store to variable year

Ask "Enter a password: " and wait for user input

Store to variable password

username = name [first 3 letters] + age

Display "Your username is", username + "."

Display "Your password is", password + "."

Display "Saving user data…"

Open connection to file ‘users.csv’ in write mode as userfile

Append username + "," + password + "," + year + "*/*n" to userfile

Wait 3 seconds

Display "Your new username is " + username + "."

Display "Your password is " + password + "."

Display "Press Enter to close."

Create new variable defined by input

Close connection to file stored in variable userfile

Exit

## Quiz

Store False in new variable loginactive

Create new empty variable answer

Create new variable qnum with a value of 0

Open connection to file ‘users.csv’ in write mode as userfile

While loginactive does not equal to True {

Ask user to enter their username

Store input in new variable username

Ask user to enter their password

Store input in new variable password

If username has matching results in userfile {

If password matches result in userfile {

Set variable loginactive to True

}

}

}

Open connection to file ‘question’+ string(num) +’.txt’ in read mode as question

Store "\*" in new variable realanswer

Comment 'realanswer equals to \* so that answer will not equal realanswer, thus preventing the quiz ending too early'

Display "Get all the answers right, OR YOU FAIL! "

Define subprocess ask () {

While answer does not equal realanswer {

If answer does not equal None {

Display "Incorrect!"

Display new line

Set qnum to qnum + 1

If qnum == 3 {

Open file “fail.jpg”

Exit program

}

Store contents of question [line 2] to variable realanswer

Display "Question " + num + ": " + question

Wait for user input

Store input in variable answer

}

}

If answer equals realanswer {

Display "Correct!"

}

Set answer to None

Set num to num + 1

If num is less than or equal to 10 {

Call subprocess ask ()  
}

Else {

Open file "win.jpg"

# Sources Table

|  |  |
| --- | --- |
| Name | Source |
| Programming Examples | http:*//*www.computing.outwood.com*/*website*/*NEA*/*programming.html |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |