*GCSE (9–1) Computer Science J276/03 Programming project – Task 1 Non-examined assessment (NEA) task June 2018 series*

**Introduction**

I am tasked to create a code that will allow students to access a quiz of 3 different subjects of 3 different difficulties. Then they will complete the quiz and their scores, percentages and grades will be written into an external text file. I will be using python to make a subject quiz and I will use procedures and validation, this will make the code more efficient and effective. I will be using resource provided by python to ensure a successful outcome for my program.

**Success Criteria**

**Criteria List:**

* The user should input the student’s name and their age and year group.
* The user should be able to understand what is printed on the screen
* My program will use procedure
* My program will use validation
* My program will be at good length of code
* My program will run
* The code has to read from an external text file
* My program will use loops, selection and menus
* My program should be able to split strings and reassemble them.

**Psuedocode/Algorithm**

Procedure Menu (Title)

Statement Introduction to the user (What is the code about)

Statement Options (Start Quiz, Debug mode, Exit) (Debug = Coder can debug issues)

Input User input (Selects options choice)

**END PROCEDURE**

Procedure First Student Input (Title) (Start Quiz = True)

Input Enter First Student’s Name (Username)

Input Enter First Student’s Age (Age)

Input Enter First Student’s Year Group (Year Group)

Output Unique Username (Splitting of Username) (Merge username and age) (E.G Zain + 16 = Zai16) (First Unique Username)

Return First Student’s Username

Return First Student’s Age

Return First Student’s Year Group

Return First Student’s Unique Username

**END PROCEDURE**

Procedure Password/Validation (Title)

Input Enter First Student’s Password (Validation Required)

Validation Longer than 7 words, has numbers, has Capitals

Return First Student’s Password

**END PROCEDURE**

Procedure Second Student Input (Title) (Start Quiz = True)

Input Enter Second Student’s Name (Username)

Input Enter Second Student’s Age (Age)

Input Enter Second Student’s Year Group (Year Group)

Output Unique Username (Splitting of Username) (Merge username and age) (E.G Dave + 16 = Dav16) (Second Unique Username)

Return Second Student’s Username

Return Second Student’s Age

Return Second Student’s Year Group

Return Second Student’s Unique Username

Return Second Unique Username

**END PROCEDURE**

Procedure Password/Validation (Title)

Input Enter Second Student’s Password (Validation Required)

Validation Longer than 7 words, has numbers, has Capitals

Return Second Student’s Password

**END PROCEDURE**

Procedure Third Student Input (Title) (Start Quiz = True)

Input Enter Third Student’s Name (Username)

Input Enter Third Student’s Age (Age)

Input Enter Third Student’s Year Group (Year Group)

Output Unique Username (Splitting of Username) (Merge username and age) (E.G Alex + 16 = Ale16) (Third Unique Username)

Return Third Student’s Username

Return Third Student’s Age

Return Third Student’s Year Group

Return Third Student’s Unique Username

Return Third Unique Username

**END PROCEDURE**

Procedure Password/Validation (Title)

Input Enter Third Student’s Password (Validation Required)

Validation Longer than 7 words, has numbers, has Capitals

Return Third Student’s Password

**END PROCEDURE**

Procedure Write all Student’s details to a file (Title)

Write Write First Student’s Name, Age, Year Group and Unique Username (Written to an external file)

Write Write Second Student’s Name, Age, Year Group and Unique Username (Written to an external file)

Write Write Third Student’s Name, Age, Year Group and Unique Username (Written to an external file)

Procedure User Select (Title)

Input User inputs what student is going first, User selects 1,2 or 3 (1 = User 1, 2 = User 2, 3 = User 3)

**END PROCEDURE**

Procedure Difficulty and Subject Selection(Title)

Input Select Difficulty of the subject (User selects E/e = Easy, M/m = Medium or H/h = Hard)

Input Select the one subject of the three. (User inputs 1,2 or 3) (1 = Computer Science) (2 = Maths) (3 = Product Design)

// (I will demonstrate one quiz for this Psuedocode as presenting nine will exceed the 20 hour limit however I will show glimpse of it in the development page. All quizzes function the same but have different questions and answers but the same score, percentage and grade system)

Statement Explain how the score work and that score, percentage and grade system will be stored in an external file)

Variable Score Counter/System (Integer Value = 0)

Variable Percentage System (Integer Value = 0)

Read Reads from the file, the first question.

Input User input 1 or 2 (1 Solution Question)

Read Reads either the correct or incorrect answer

Variable User Either Gain or Lose points

Variable User's Percentage either increase or decreases

Read Reads from the file, the second question.

Input User input 1 or 2 (1 Solution Question)

Read Reads either the correct or incorrect answer

Variable User Either Gain or Lose points

Variable User's Percentage either increase or decreases

Read Reads from the file, the third question.

Input User input 1 or 2 (1 Solution Question)

Read Reads either the correct or incorrect answer

Variable User Either Gain or Lose points

Variable User's Percentage either increase or decreases

Read Reads from the file, the fourth question.

Input User input 1 or 2 (1 Solution Question)

Read Reads either the correct or incorrect answer

Variable User Either Gain or Lose points

Variable User's Percentage either increase or decreases

Read Reads from the file, the fifth question.

Input User input 1 or 2 (1 Solution Question)

Read Reads either the correct or incorrect answer

Variable User Either Gain or Lose points

Variable User's Percentage either increase or decreases

Selection Condition If score is certain value, the user either get A,B or C (Depending on the score not percentage)

Write User's Scores, Percentage and Grades is written to an external file

// (This repeats through all the easy quiz and the medium and hard gain an extra answer) (The Student can only view his/her score through an external file)

**END PROCEDURE**

Procedure Debug Mode (Title)

Input Admin inputs his password

Validation The password has to be a certain string to verify and move on. If incorrect it will return to menu.

Input The Admin input a quiz choice and get transferred to the selected quiz.

**END PROCEDURE**

Exit If user presses 3 the code exits

**Justification of approach**

I'm using a Procedures because, it much more efficient and easier (in my opinion) to code couple of line you want to reuse again and to assign it to a variables for quick and easy use. Procedures are the main parts of the code to make it run.

I'm using a For-loop because, I can use it to function as a writing text one character at a time, to give it aesthetic and high level look at what python can offer. The For-loop used here is not for looping question or statement when either condition is seen with a looped amount but used as a tool rather than sequence or iteration.

I'm using a While-loop because, the while loop is used mostly for validation and uses less selection. This is because, I want the loop to happen if a user writes in an incorrect value or character which may cause the code to have an error. Having a While-loop will stop certain errors as it will loop over a piece of code reminding the user it should be like this rather than that. It could be seen as looping until a certain condition is met but rather it is used to validate the user on any false accusations.

I need to use external text files as I require to save user's details and scores into external text files.

I need to use a print statement to print out my code and be presented on the shell.

I could of used lists to help the code validate or be used as a "if decision" so that the code would find the that certain word from the array.

// Note to examiner: I do use lists as suggested above as I thought of ways to add more validation and variety to my code

**Variables**

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Variable Use |
| student\_name | String | Used to store in an external text file and used in the unique username. |
| student\_age | String | Used to store in an external text file and used in the unique username. |
| student\_year\_group | String | Used to store in an external text file. |
| student\_unique\_username | String | Used to recall the string merged of username and age, to be used in multiple procedures. |
| student\_password | String | Used to store in an external text file. |
| score\_system | Integer | Used as counter to increment increasing points or decreasing points. |
| percentage\_system | Integer | Used as counter to increment increasing percentages or decreasing percentages. |
| grade\_system | String | Used to give a grade to the student depending on their score |
| admin\_password | string | Used to validate the admin of the code to access debug mode. |
| validation\_variable | Boolean | Used as validation to either allow or deny certain conditions or inputs |

**Validation**

* My program will be case sensitive and allow the user to input in lower case or upper case and will not be using .lower() and .upper()
* The program will not consist of the following validation functions: .title(), .isdigit() and any()
* My codes validation will be consisting of a module know as RE (Regular Expression) which will verify certain character in the string and will validate if they are their or not
* The code will validation to check lengths of stings and use while loops as Boolean to see if the value meets the condition and if it true or false. True will move on and be valid while False is the invalid option.

**Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test** | **Success Criteria** | **Test Data** | **Expected Outcome** |
| 1 | Test that my program will run without errors or crashes | My program will run without errors or crashes | **Normal Data:**  Zain 1 2 3 4  **Erroneous Data:**  Zain 1  Syntax error  '2' is not defined | **Normal Data:**  The Program runs with no errors or crashes  **Erroneous Data:**  The Program outputs a syntax error because, the programmer forgot to make 2 a string and instead made it a variable within input statement |
| 2 | Test that my program will ask the user to input a sentence | My program will ask the user to input a sentence | **Normal Data:**  Enter your name:  **Erroneous Data:**  N/A | **Normal Data:**  The Program outputted a sentence and allows the user to input a sentence  **Erroneous Data:**  The Program doesn't allow the user to input the sentence because, there is no input function in the program |
| 3 | Test that my program can store user inputted sentences into a list | My program will ask the user a sentence and that sentence will be appended to a list | **Normal Data:**  Name:  Names: Zain Bob  **Erroneous Data:**  Name:  Names: Zain N/A | **Normal Data:**  The Program allowed the user to input a sentence and it was appended to the list.  **Erroneous Data:**  The Program allowed the user to input a sentence but was not appended to the list due to the user inputting a digit instead of a word |
| 4 | Test that my program can recognise when the user inputs a digit or a letter | My program can identify the difference between a string and a integer | **Normal Data:**  Zain,12  **Erroneous Data:**  1Zain2 | **Normal Data:**  The Program was able to identify that 'Zain' is a string and '12' is a integer.  **Erroneous Data:**  The Program wasn't able to identify that user inputted the string as name and the integer as a number. It believed the number is the string. |

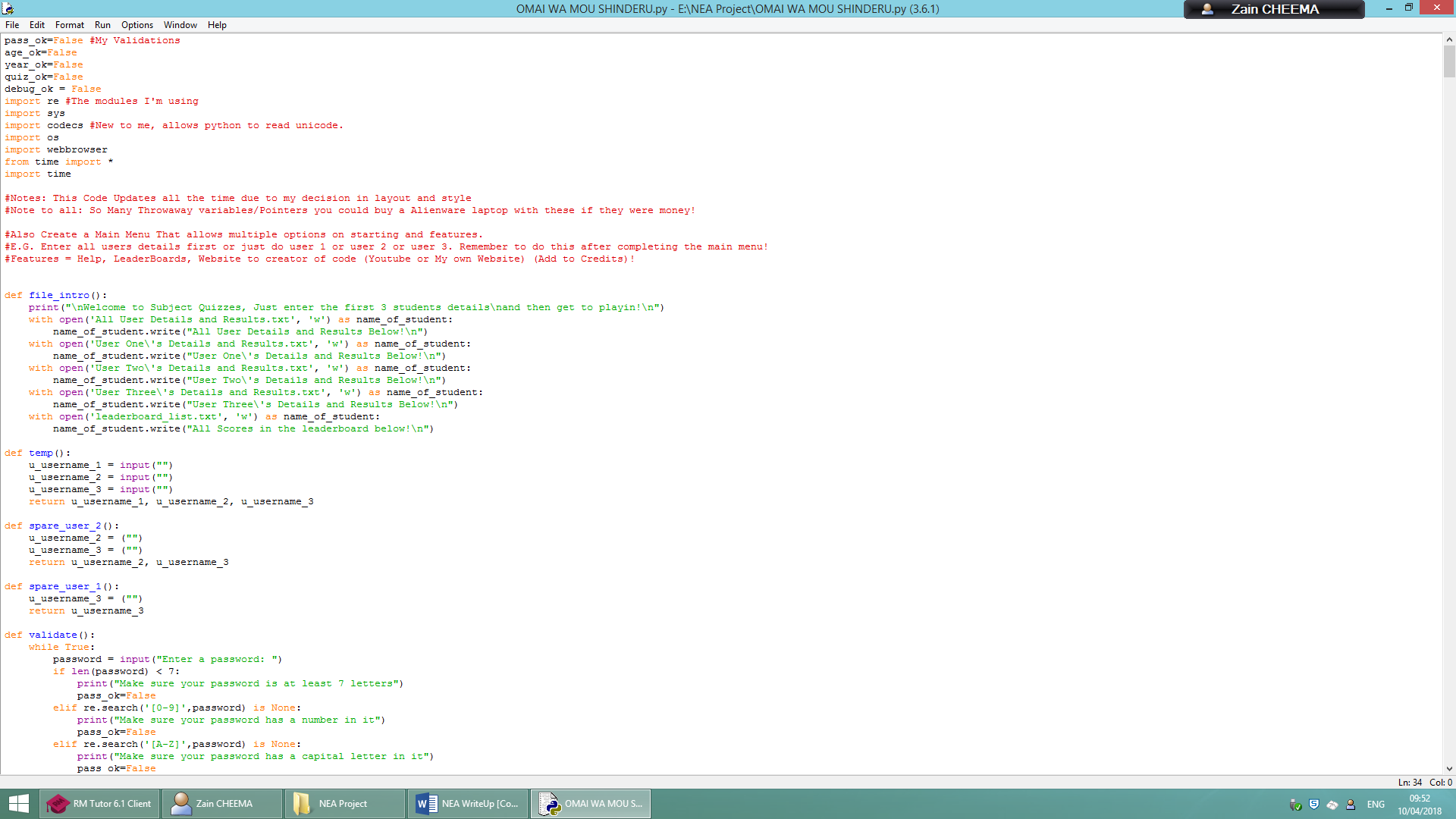
Development on next page…

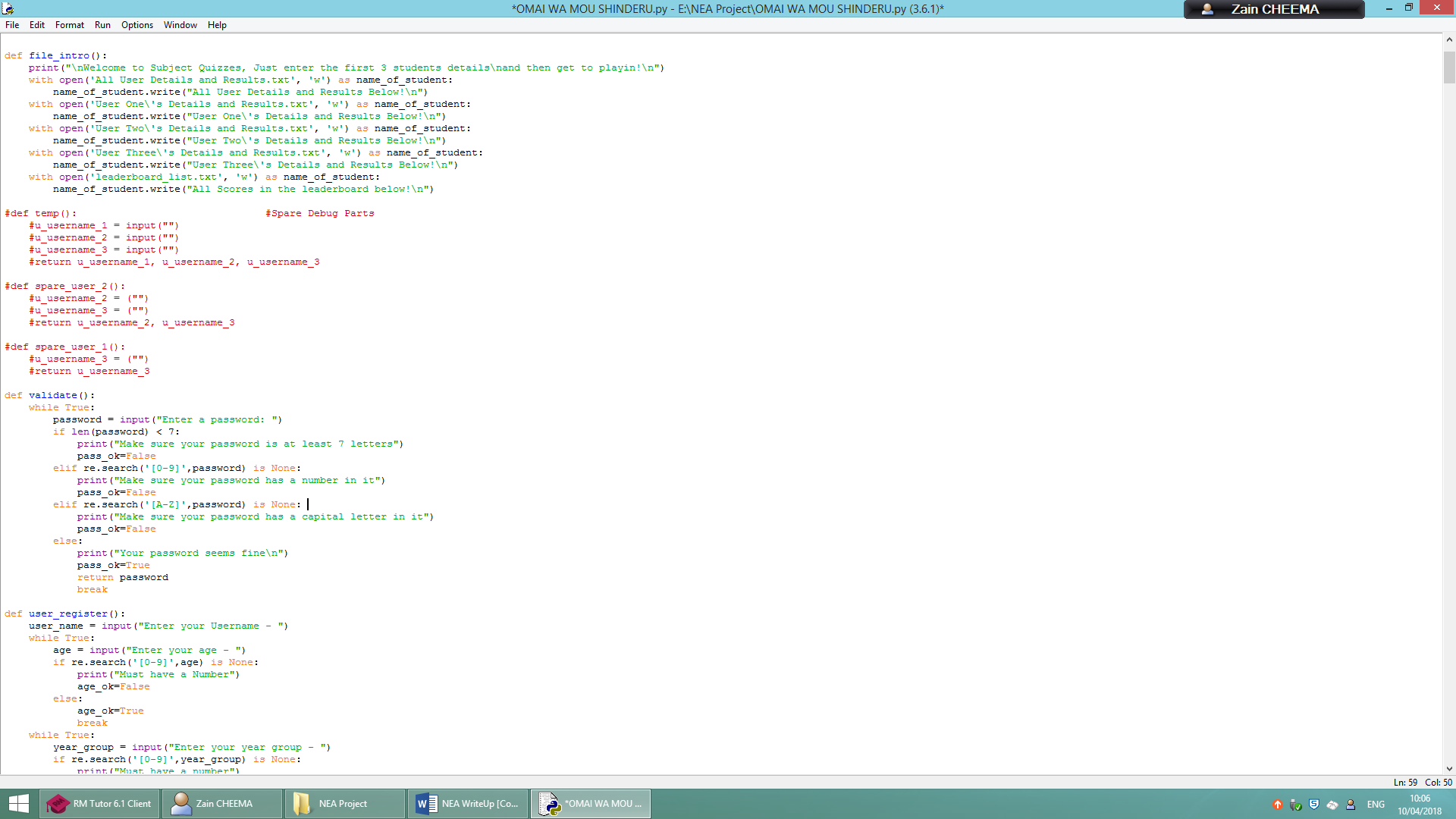
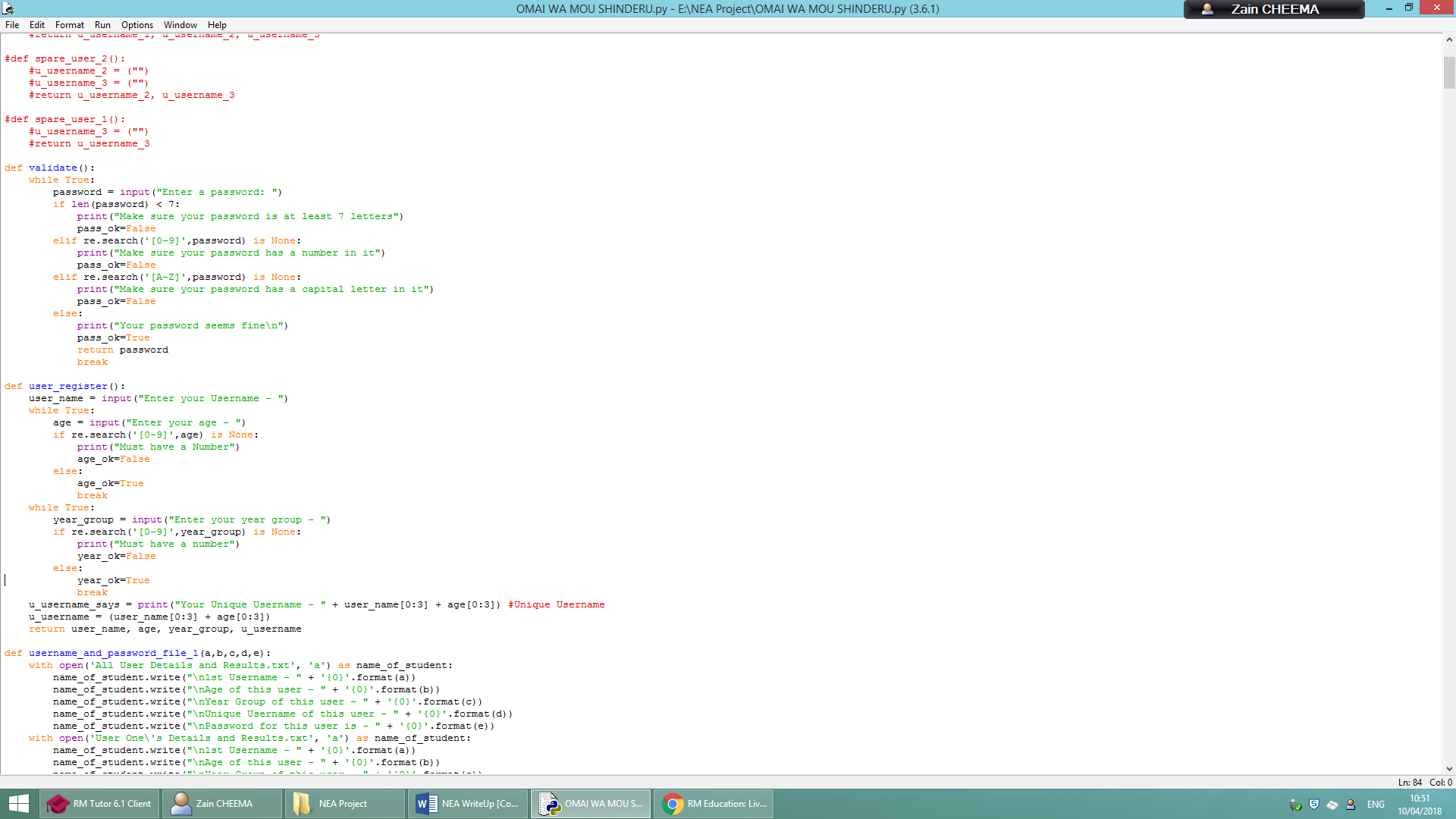
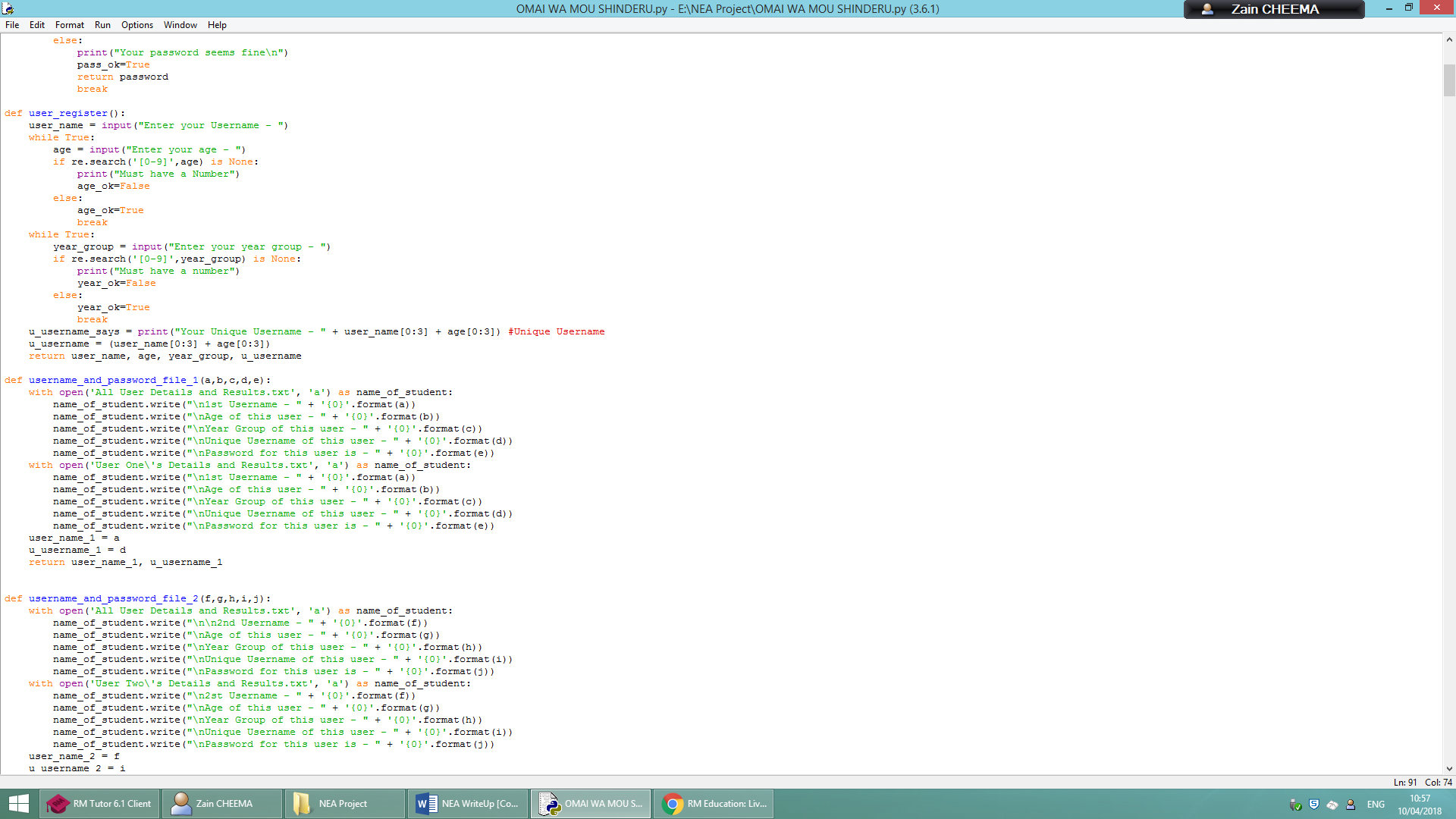
**Development**

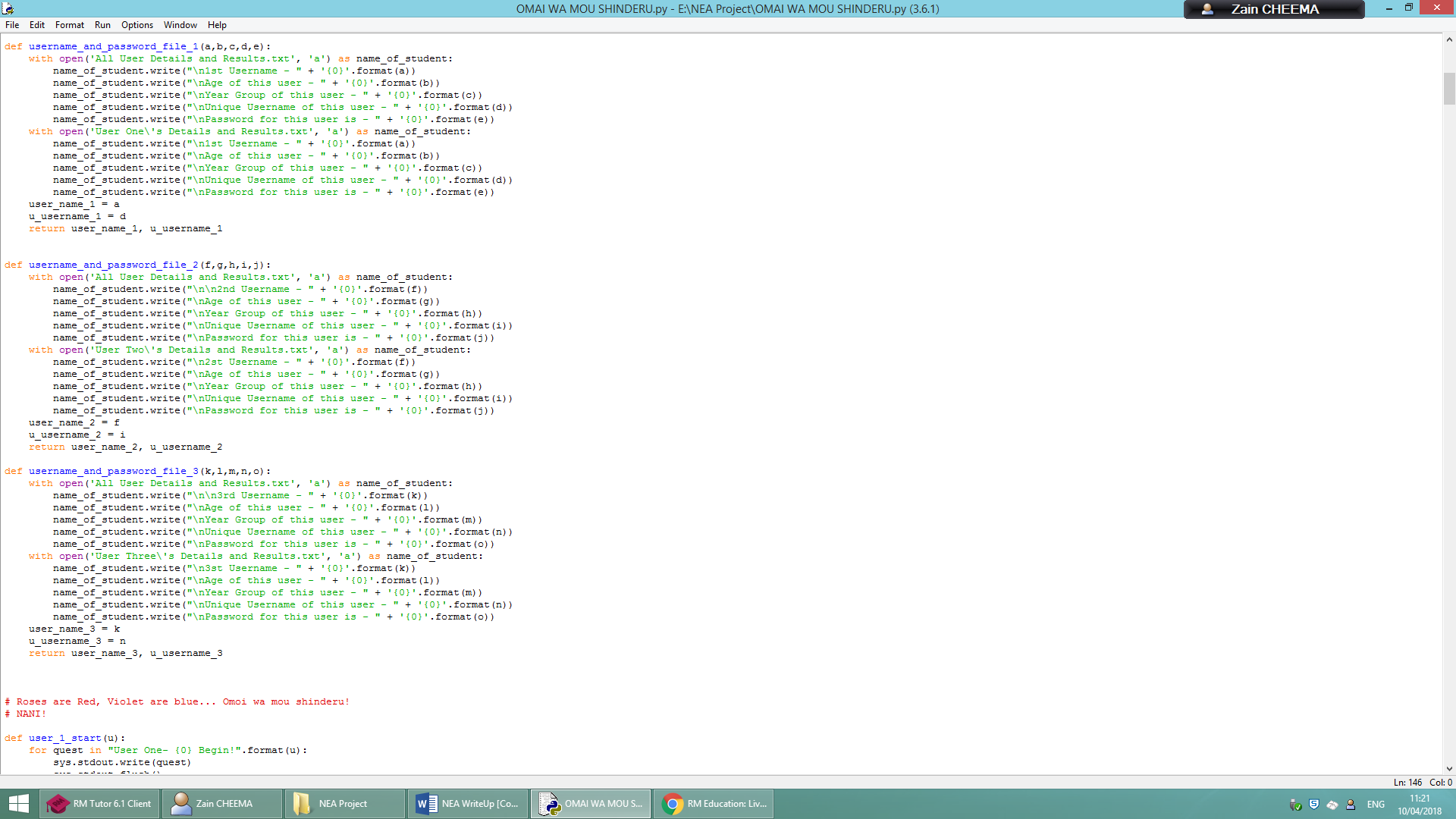
**// Note to examiner: This page is not in true order on how the code will be displayed to the user but rather in the order and worked on it. Also this code shown is the final parts of my code.**

1. These are modules and validation variables that are used in my code.

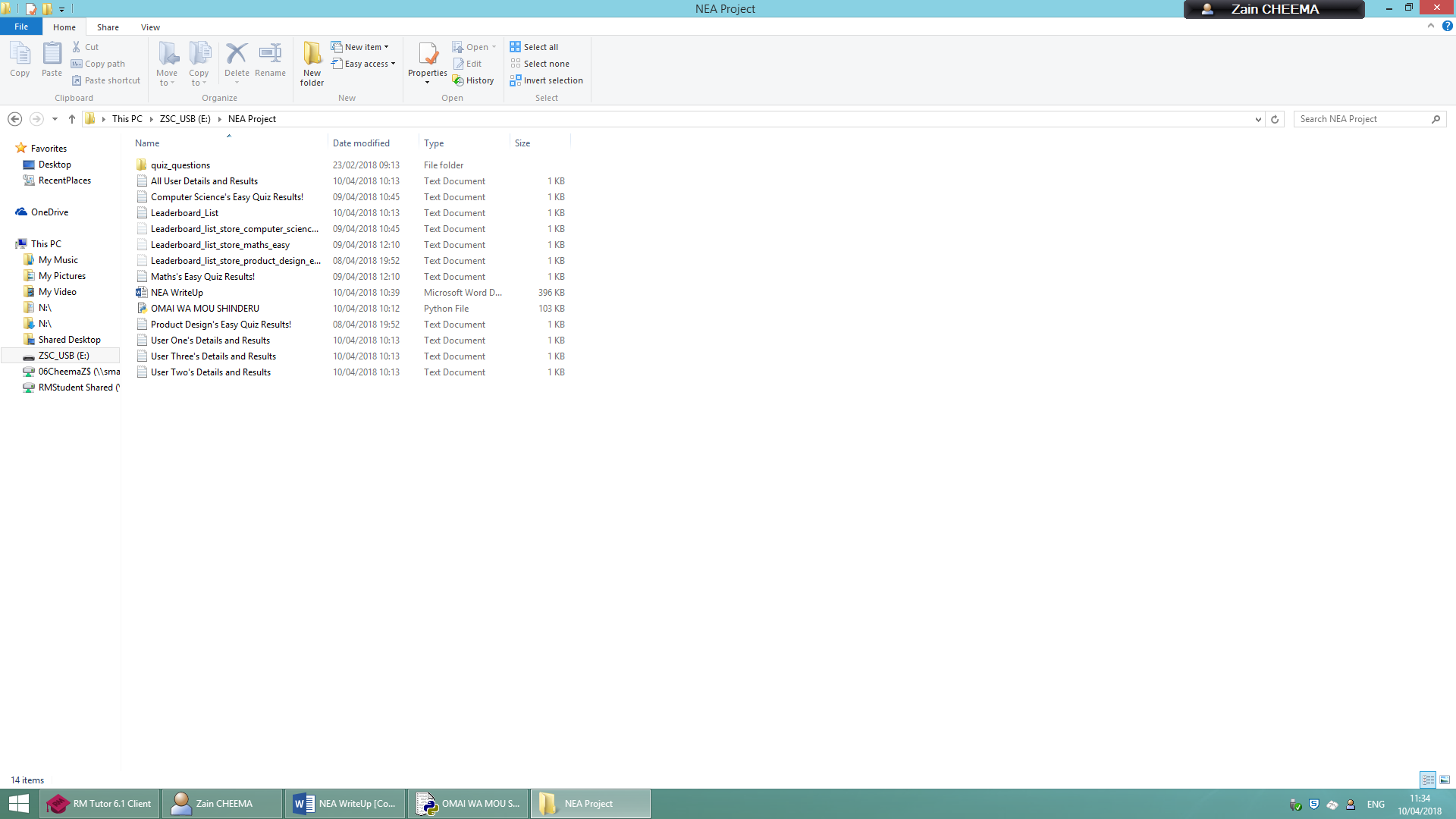
pass\_ok, age\_ok, year\_ok, quiz\_ok, debug\_ok – My Validation Variables

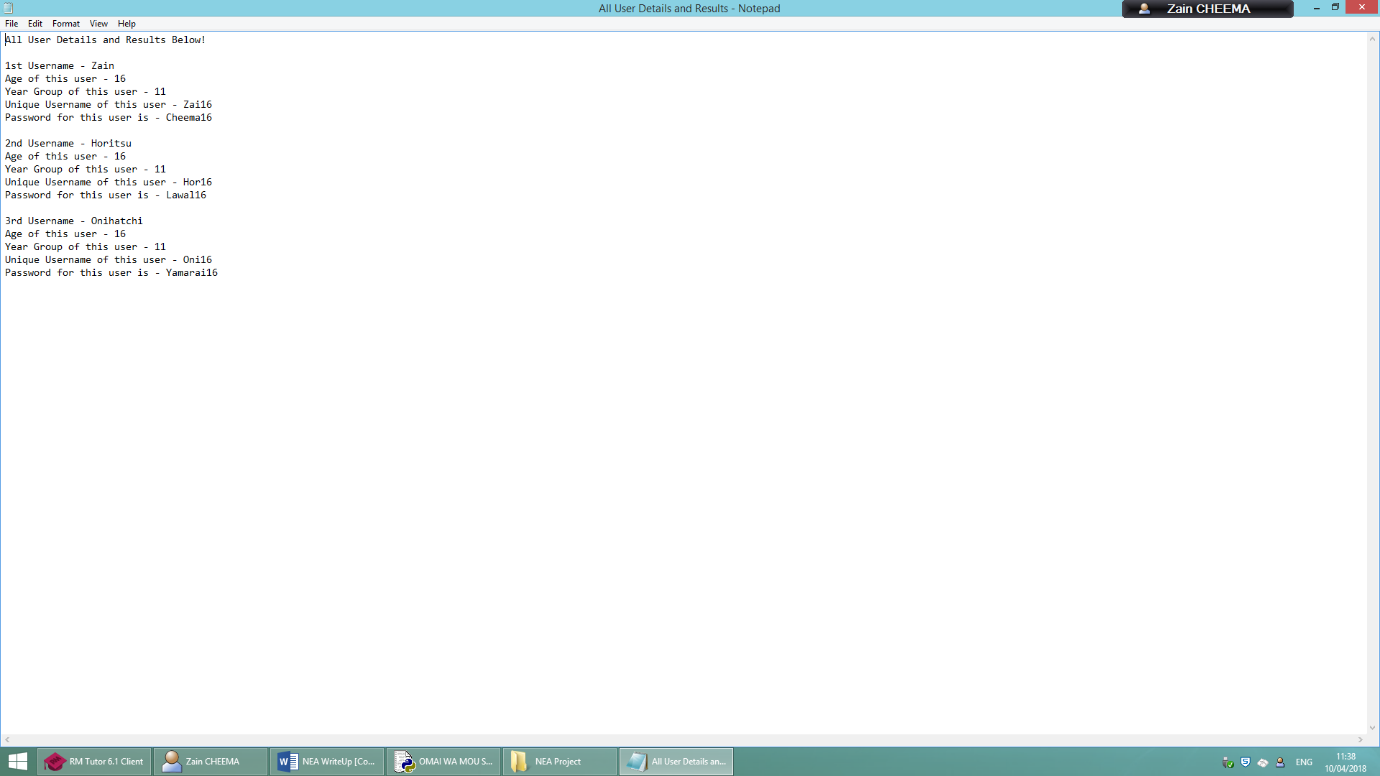
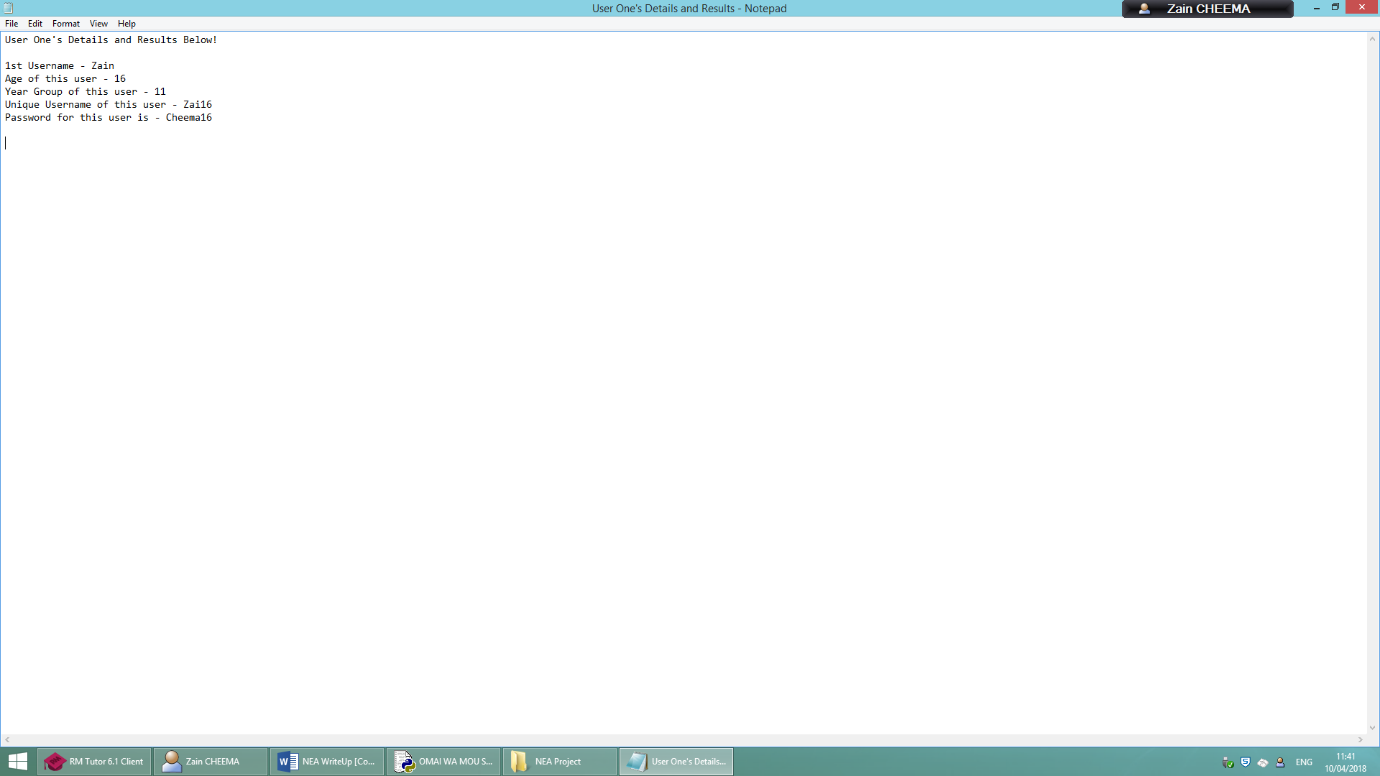
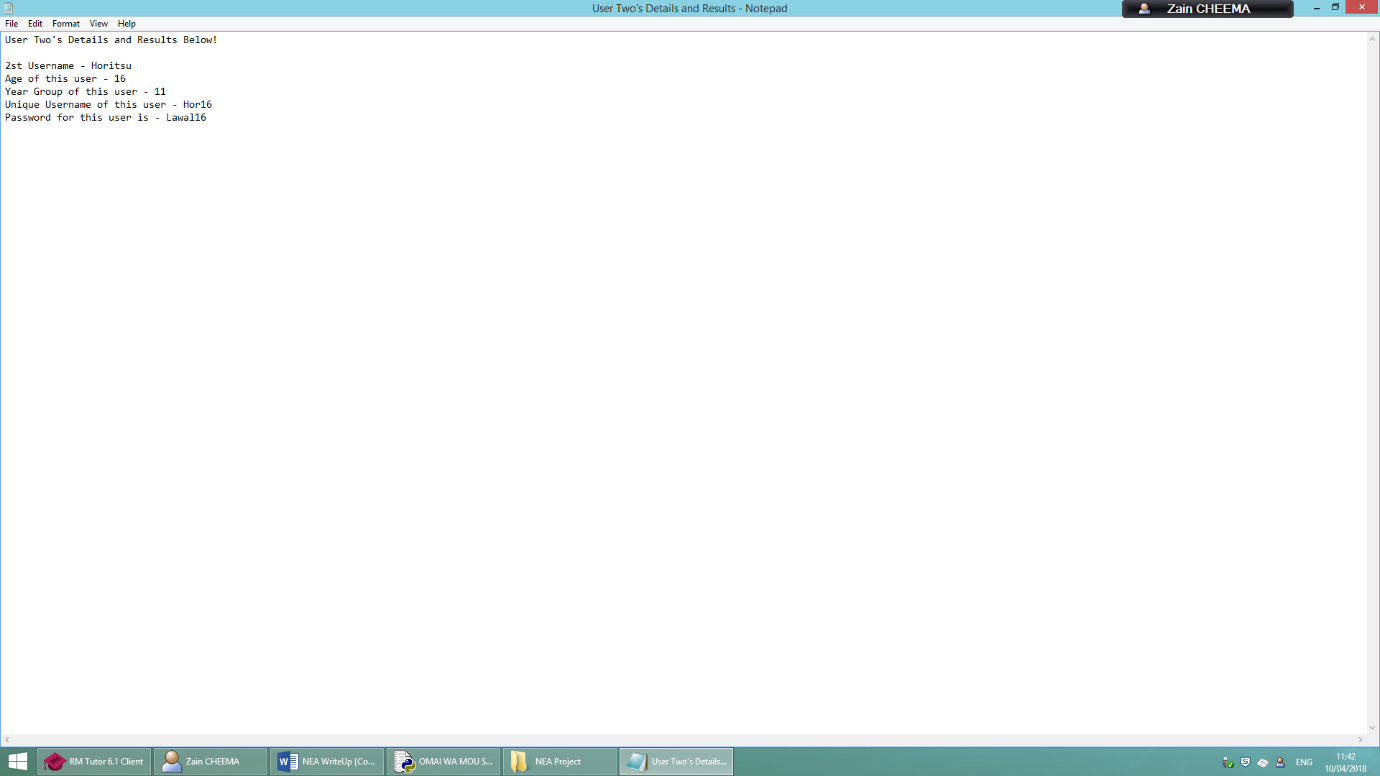
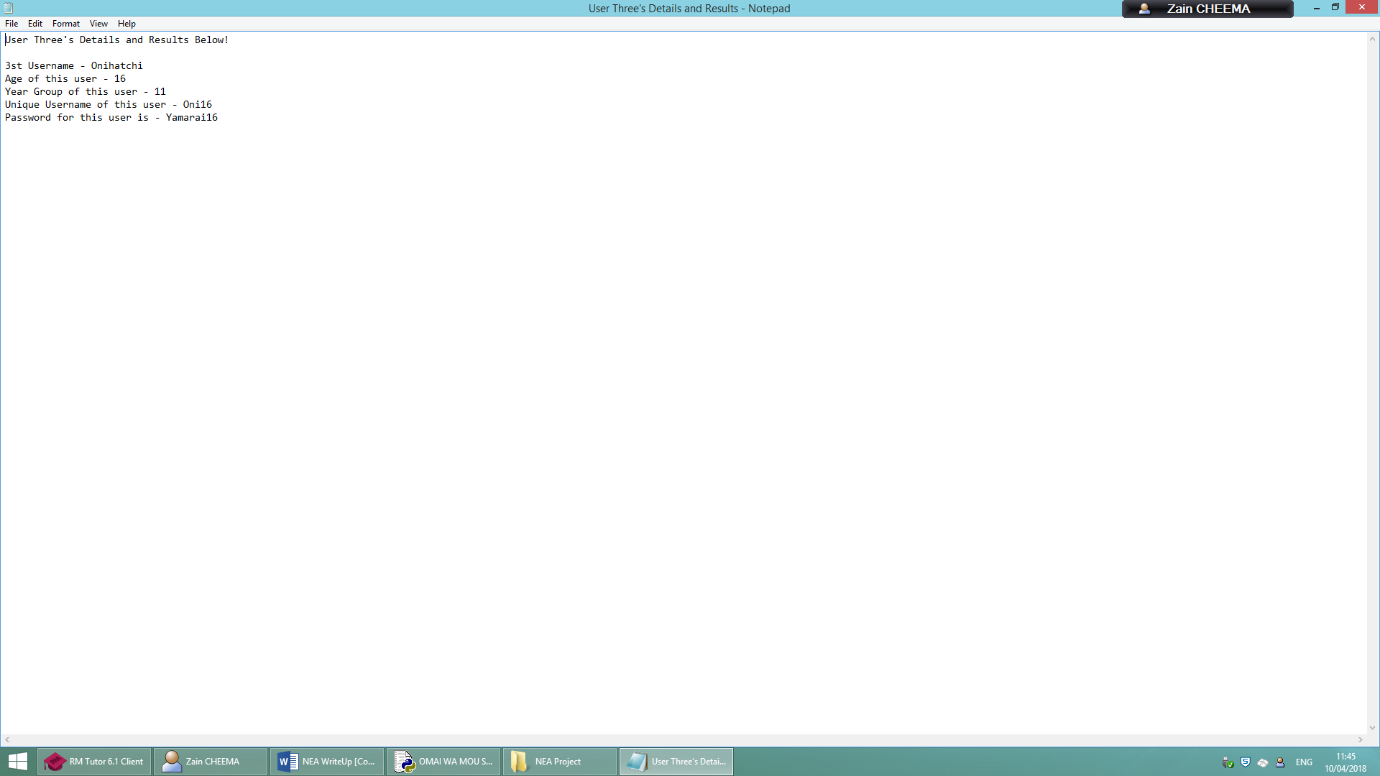
re, sys, codecs, os, webbrowser, time, \* from time – My Modules

1. Below is how the details about each user and both users are being going to be written into a text file. This procedure “file\_intro()”, will create the 5 separate text files. The last text file is for a leader board system I developed for competition among students. To view your highest score in the leader board text file.
2. Below is the password validation. The procedure “validate()” is separate from the procedure “user\_register()” (user\_register() is explained below) because both procedure will correlate and will work with each in different scenario e.g. into text files. The line “password = input(“Enter a password: “)” is the line where user inputs the password for their account and the code will then use the module RE in the function re.search to validate things like Integers and Capitals, the code would validate any missing parts by notifying the user that they are missing a integer or capital character. The len function is verify the length password and if the length is invalid it requires the user to make the length longer. Then if it finds no issues then it will carry on, notify the user and return password. Then end the loop.
3. Below is the procedure “user\_register()”, this procedure asks the user of their name, age and year group and at the end it concatenates the string and 2 variables which split character and combined them to make whole new output! The variable “user\_name” will ask the user to input their name or username, the same thing. It won’t need validation as if a name as integer or float in it then that is possibly the name they were given at birth for example, za1n. Then age and year group is asked I add an integer validation so that if they were to make a mistake then the code will notify them and they would fix it. Finally the code prints to the user their “unique username” and the variable underneath is not need as I could of done that above then concatenated the variable “u\_username” to “u\_username\_says” but that just a way to uplevel it. Returns all the variables from this procedure for later use.
4. Below is 3 procedures that will write all the information into 4 different external text files which will be shown on step 6. The code below opens the text file and appends each variable to a line of the .txt file. It does not write each one because writing it will over write previous data and the best way to fix that is in procedure “file\_intro()” I have a write function working there so that every time I reset the code it will erase the last users data from the txt file to allow the newly append files to be written in it. You can also see that the code has new variables that are appended to throwaway variables because this variables will be returned into new sections of the code.

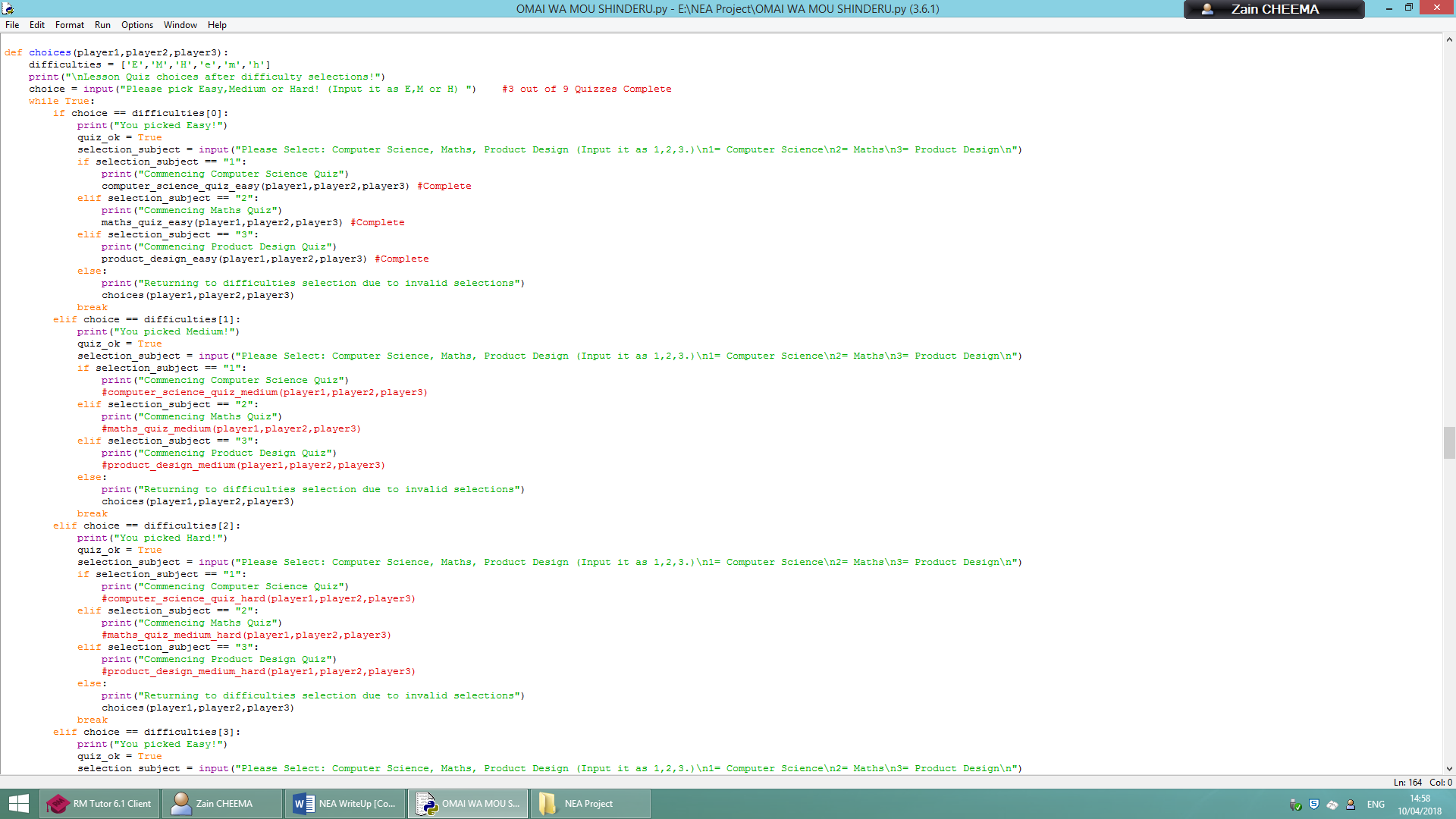


1. Below shows the files written into text files within the folder of the code. The text file below are from, “All User Details and Results”, “User One’s Details and Results”, “User Two’s Details and Results” and “User Three’s Details and Results”. The files reset every time I rerun the code.

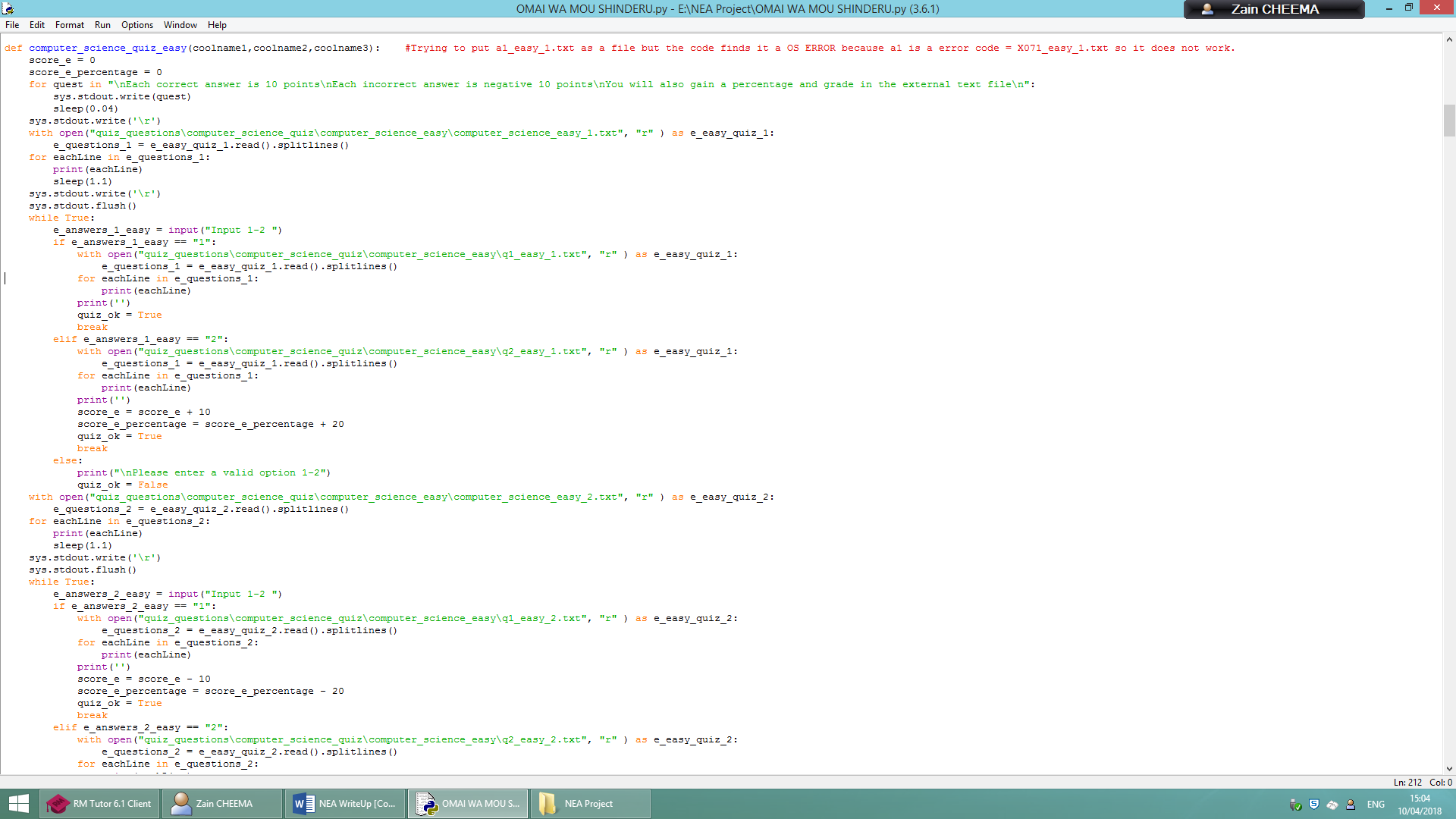
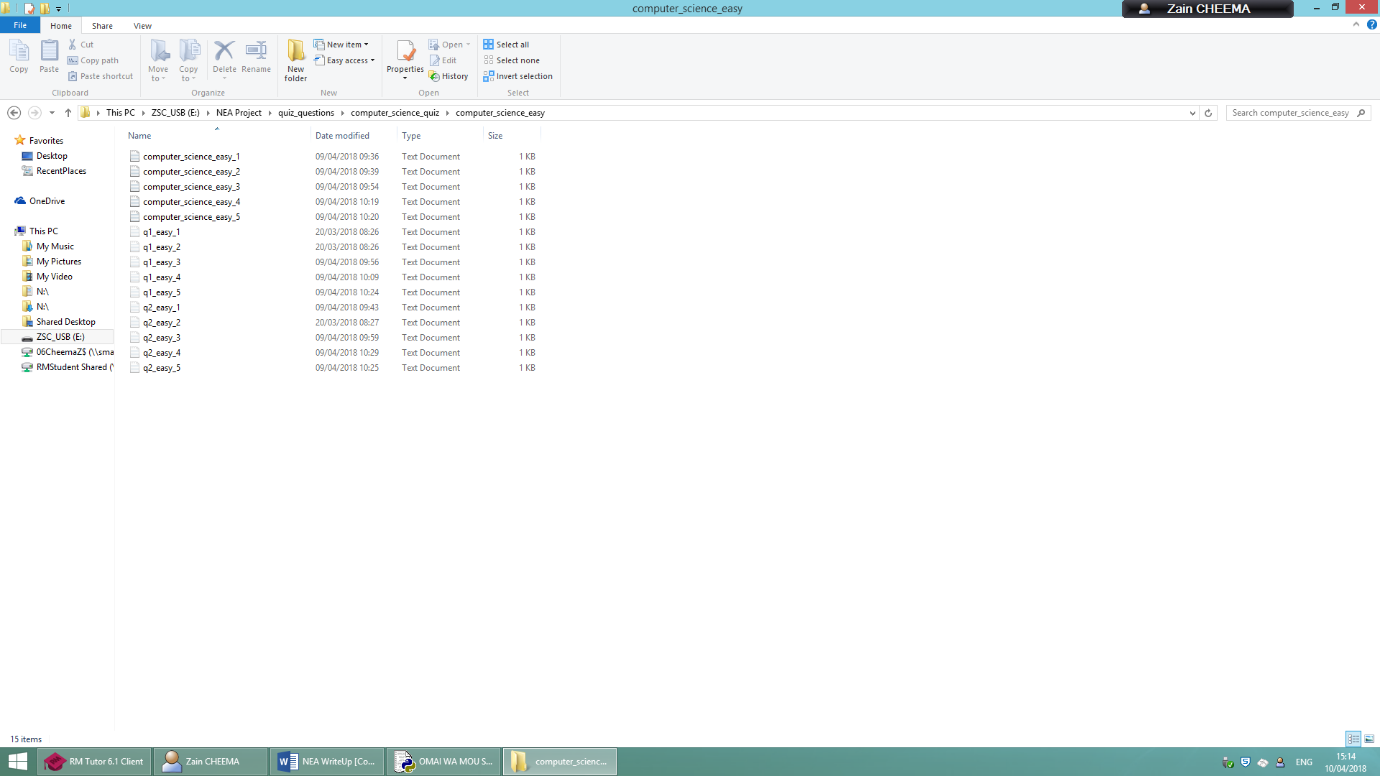


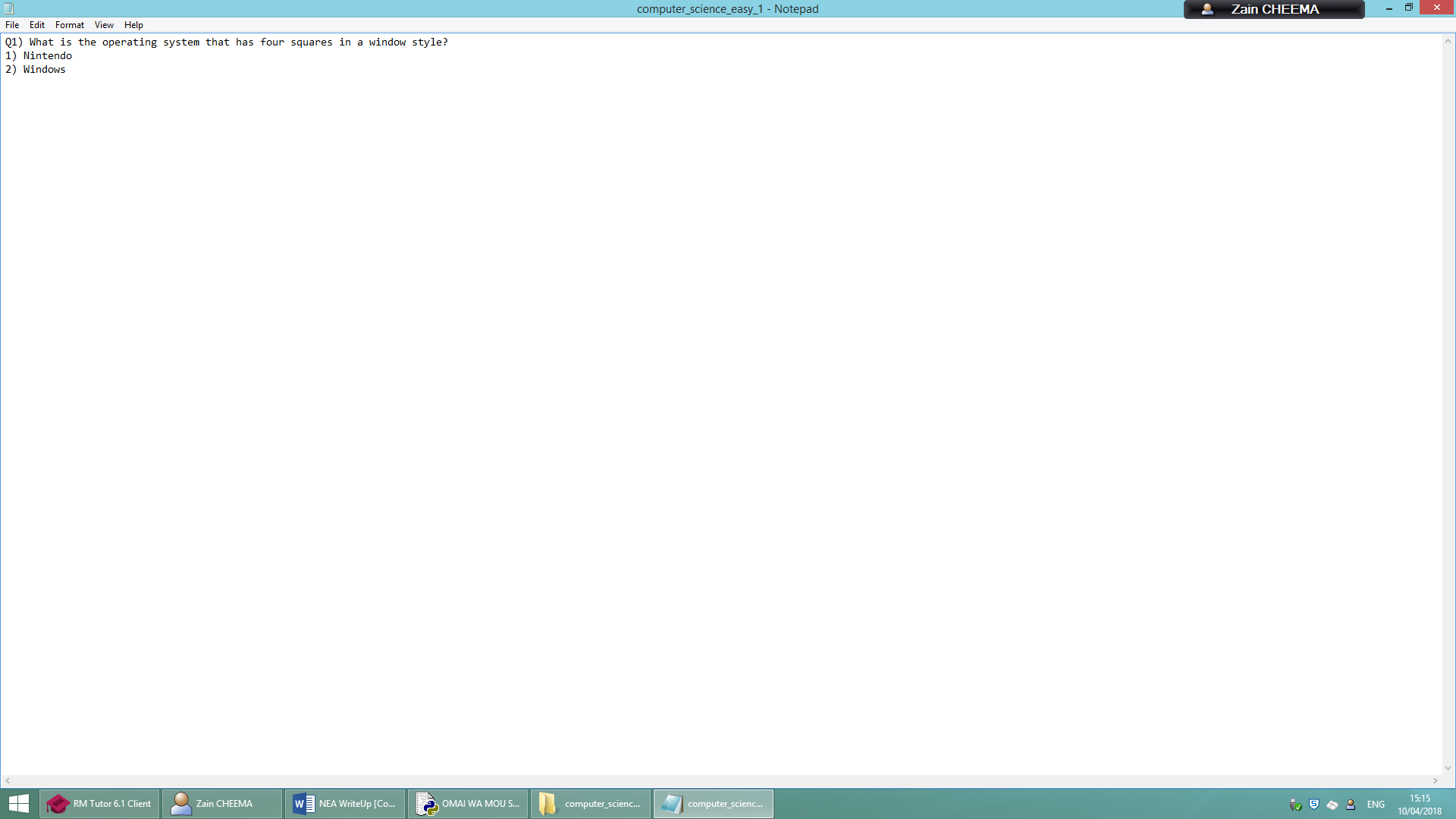
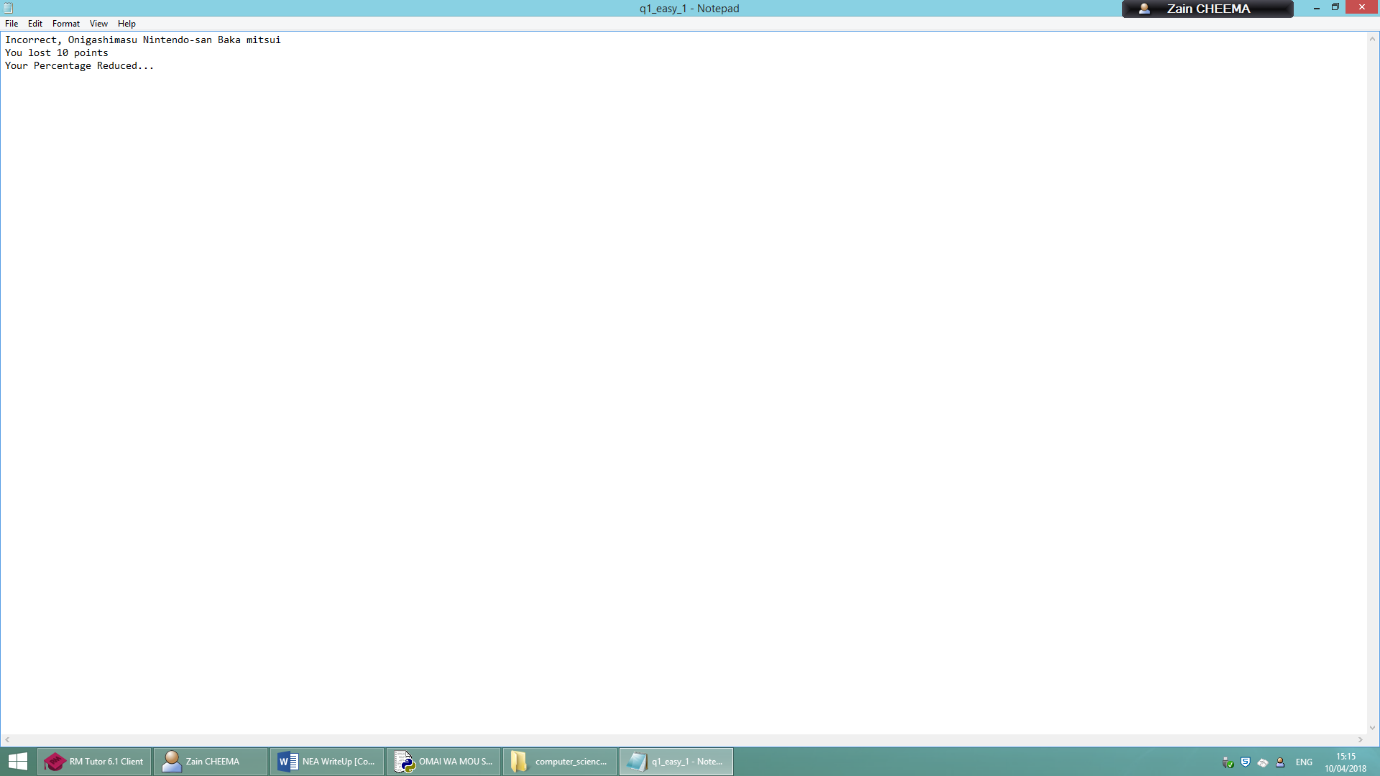


1. The Procedures “user\_1\_start(u)”, user\_2\_start(xz) and user\_3\_start (xy) basically are procedures that happen after procedure, “places(number1,number2,number3). The “places(number1,number2,number3)” procedure allows the users to select which user goes first. The variable “who\_first” allows user to decide who to go first by putting 1,2 or 3. Then the other procedures will allow that certain users name to be said, for example “User One – Zai16 Begin!”. Then goes to the game menu (aka subject menu)
2. Below shows the procedure called “choices(player1,player2,player3)”, this procedure is the main quiz procedure and it allows the user to select there difficulty from a user input variable called choice. This then after go to one of the “if selection” choices depending of difficulty and case sensitivity. Then you select ether computer science, maths or product design by inputting 1,2 or 3. This then let’s the code run the procedure that goes to one of the quiz codes.



1. Below is the procedure of “computer\_science\_easy\_quiz(coolname1,coolname2,coolname3)” and this is one of my nine quizzes. It has a score and percentage variable with the starting integer value as zero and the value either increase or decreasing depending on the answer selected by the user. The external text file reads the questions from the file location: “quiz\_questions\computer\_science\_quiz\computer\_science\_easy”

The answer also are located from this area but the files are hidden in plain view so that the user cannot access them to cheat from them or configuration of these files. So they are hidden files. When they answer the code will open the file location of the answers from the folder and establish to user if they gain or lose points and what happens to the percentage. Then the percentage is sent to a “if selection” that decides on their grade and sends all this to a file called “Computer Science’s Easy Quiz Results”



//Note to examiner, I only have a 30 min left so I cannot complete full analysis on development so I move onto evaluation

**Evaluation**

In this NEA project, I managed to achieve all of the success criteria in an efficient way. My program also ran with crashing

* My Program ran without crashing
* The user was able to input data and get it outputted clearly
* The program is at a good length
* The Programs Validations have worked

Also I was overloaded with many interesting errors like an error saying I had to convert one of my files to Unicode because I used Japanese and an overflow error for making it write to multiple files.

Time Taken

* Write-Up: 6 hours
* First Code: 14 hours
* Total: 20 Hours