**Some Dependencies of this automation script**

This script is written in python 3. Python is an interpreted language, therefore to ensure that this script runs on your computer effectively, you need to install Python 3 on your computer.

Download and install python 3 onto your computer here:

<https://www.python.org/downloads/release/python-383/>

I recommend you download the executable version for windows X86 or X64 depending on your system’s requirements.

This script is intended to be run from the command line preferably Windows Power Shell. Also, the script, the excel file, the template word document and the template letter, all need to be located in the same folder as the script is written to work with files in the current working directory.

This script also uses some third party python user modules, which include;

**openpyxl.** The version of openpyxl used is 2.6.2 and you install it by running the following command on windows power shell/command prompt: **pip install --user –U openpyxl==2.6.2.** Ensure you have internet connection on your computer when you run this command. Also Windows PowerShell should be run as administrator for this command to be effectively run.

**python-docx:** the version of python-docx being used is 0.8.10 and you install it by running the following command on windows power shell/command prompt **pip install -user -U python-docx==0.8.10.**

**How to run this script**

Launch Windows PowerShell/command prompt as an administrator by going to your start menu, type in power shell in the search bar, right click on Windows PowerShell desktop app and click run as administrator.

When the PowerShell interface is open you need to change the current working directory to wherever on your computer this script is located using the change directory command. Type the following into your command line:

**cd C:\users\(computer name)\Downloads\wordAutomationProject**

Please note that the directory specified above is only a skeletal structure. You will have to specify the directory depending on where the project is stored on your computer. Therefore, the only independent part of the above directive, is the command; ‘**cd’**.

The final step is to run this script using the following command:

python automateResultLetter.py **<name of excel file>.**xlsx **<name of letter template>**.docx

This script can also be run on command prompt using the same commands.

**NOTE!!**

* The file names of the excel template and word letter template files, in this case; “sampledata.xlsx” and “SampleLetter.docx” should not contain a space character. You can separate two distinct words by using an underscore in the file name.
* You can run this script on as many excel files containing students results as possible provided the rows and columns are arranged in the same way as the sample file shown here.
* Only the excel result file can be changed, the files named; **“SampleLetter.docx”** and **“result letter”**, as they contain the text and formatting needed to produce the output result letter. Therefore, this script can be used in its exact format only by those who use the same colour grading system of Green, Amber and Red.
* It is important to note that the main script; “automateResultLetter.py” is dependent on some functions defined in the accompanying script; “writeDocx.py” and therefore both should be present in the same folder as the required files.
* It is also worth noting that this script produces two different documents as output which will both be saved in the same folder as; “student letters\_<num>.txt” and “student letters\_<num>.docx”. These are the output files and you will likely be more interested in the .docx format of the output as the .txt format is only an intermediate result of the script running process. <num> can be any integer, depending on how many times you have used this script on your local computer.