Assignment 1

Student Name: Eryk Gloginski **(L00157413)**

Course: BSc Computing

Module: Automation

Lecturer: Saim Ghafoor

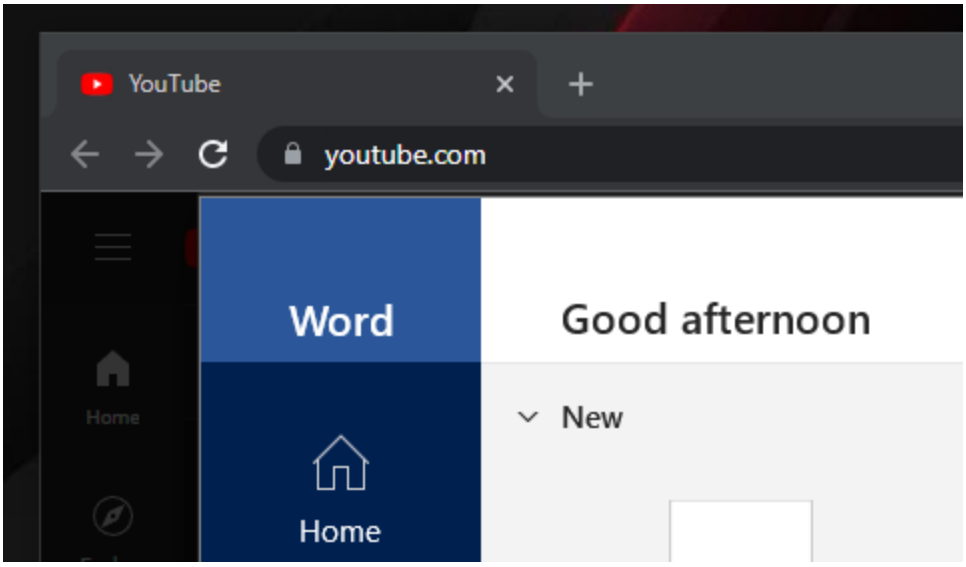
Submission Date: 4/01/2022

Question 1:

Part 1: First, I make sure that only the output is visible using the **"@echo off**" and "**cls**" to clear the window. It is a simple program to start 2 applications on the machine that we are running the script on, starts **Google Chrome** on the **YouTube** website in incognito and starts **Windows Microsoft Word**.

```
:: make output visible only and clear screen right after
@echo off
cls
echo "Part 1: "
:: start google chrome in incognito on the youtube website
start chrome www.youtube.com /incognito
:: start microsoft word
start winword
echo "Whenever ready, press any button to continue. "
pause
```

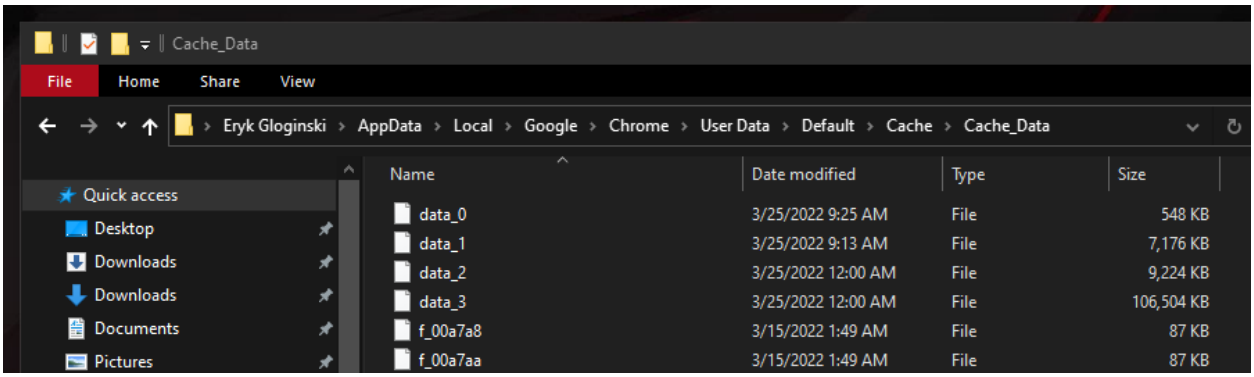
The output of the script is below:

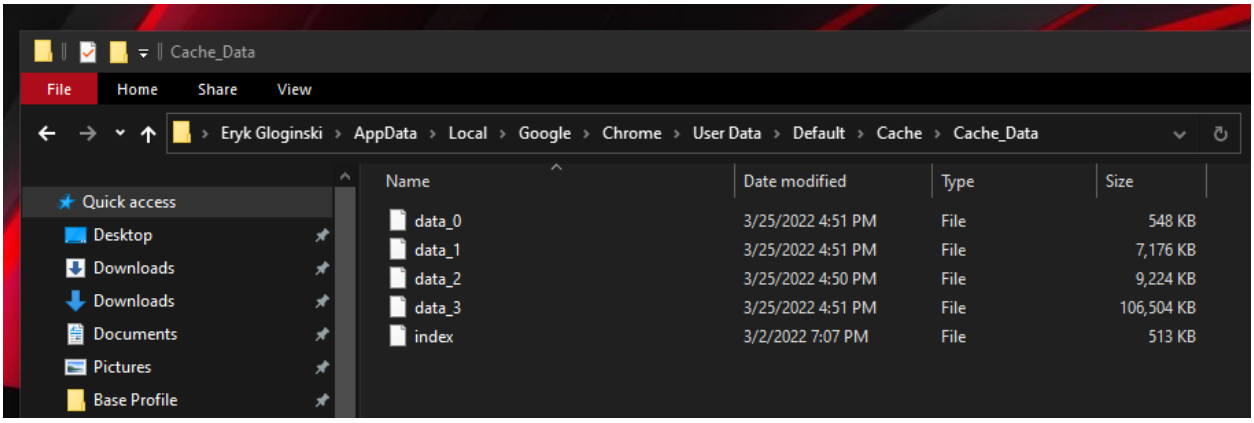


Part 2: I simply try remove the **Google Chrome Cache_Data** folder with all the files cache inside where it will leave a few important(current session) files behind.

```
echo "Part 2"
:: delete everything in the directory without prompting for confirmation
RMDIR /s /q "C:\Users\%USERNAME%\AppData\Local\Google\Chrome\User Data\Default\Cache\Cache_Data"
echo "Whenever ready, press any button to continue. "
pause
```

Below is the **before** and **after** of running this script:





Part 3: Not complete due to the complexity.

Question 2: This is the **python** script, if you are planning on running it from **CLI** or from **VSCode**, please make sure to be in the exact folder of the **.py** file. Everything is also commented.

Part 1: Firstly, I declare a method called **password_check** which returns a boolean to check if the password is valid.

```python
# PART 1 METHOD
def password_check(password):
    # declare instantly true and change it if something is not present
    val = True
    # if any letter is a digit
    if not any(letter.isdigit() for letter in password):
        val = False
        print("No Numbers in password! ")
    # if any letter is an upper character
    if not any(letter.isupper() for letter in password):
        val = False
        print("No Capital Letters in password! ")
    # if any letter is a special character
    if not any(not letter.isalnum() for letter in password):
        val = False
        print("No Symbols in password! ")
    # return boolean value
    return val
```

Then, I take in the required variables such as the **first name**, **last name**, **username**, **email id** and **password**. After that, I use a while loop to keep looping over the password check until the user enters a valid password. Finally, I save the details collected in this part into a file called "**data.txt**" where I save the info in a "**fname,lname,emailID,username,password**" format.

```python
# PART 1
# ask for whatever required
fname = input(">Enter your first name: ")
lname = input(">Enter your last name: ")
username = input(">Enter your username: ")
emailID = input(">Enter email ID: ")
pw = input(">Input password: ")

# while loop to check if password is good enough
while (password_check(pw) != True):
    pw = input(">Input password: ")
# save password into txt file with username on a brand new line
with open("data.txt", "a+") as f:
    #create password string and then append it to file
    text = fname + "," + lname + "," + emailID + "," + username + "," + pw + "\n"
    f.write(text)
```

This is the first part of the code running:

```
>Enter your first name: saim
>Enter your last name: saim
>Enter your username: saim123
>Enter email ID: 777
>Input password: asd
No Numbers in password!
No Capital Letters in password!
No Symbols in password!
>Input password: 123asd
No Capital Letters in password!
No Symbols in password!
>Input password: Asd123
No Symbols in password!
>Input password: 43@1saim
No Capital Letters in password!
>Input password: 43@1Saim
```

Part 2: This is the more difficult part where I **must** let the user login. Since we are working with a **timer** this time, I have imported the **time library** into the program. I set the **attempts** to 3 tries. I take input for the **username** which will be called **loginName** and I make a list to save the "**data.text**" lines into. I open the "**data.txt**" file to read it, I save the lines without the "**\n**" character and then check if the **loginName** is in the list. If it is, save that exact line as a **loginString**.

```
# PART 2
# declare attempts variable, ask for login and declare lines list/array
attempts = 3
loginName = input(">Enter your Login: ")
lines = []

# open data.txt in read only format
with open("data.txt", "r") as f:
    # take each line without newline character into lines list/array
    lines = [line.rstrip("\n") for line in f]

# for loop to check if the loginName is in lines list/array
for line in lines:
    # if it is, take that line as loginString
    if loginName in line:
        loginString = line
```

I split the **loginString** using the "**,**" separator into another list where I can use that list to take the 4th element of the list and save it as the **loginPassData** that was saved from the "**data.txt**" lines. Then I make a while loop to loop over the attempts, I take the **loginPassword** and compare it with the **loginPassData**, if they are the same, I inform the user that login has been **successful**. If they are not the same, inform the user that login was **not successful** and **decrement** attempts. Finally, if the user reaches 0 **attempts**, I inform the user that they have run out of attempts and that they must wait **2** minutes. The

timer that I have mentioned earlier comes in here, I put the program to sleep for **120** seconds. Once that **timer** passes, I restore the **attempts** to 3 and let the program loop again.

```python
# split loginString into separate list/array
partsOfLoginData = loginString.split(",")
# and take last part of LoginData as it's the password
loginPassData = partsOfLoginData[4]

# while loop to loop over while attempts are not 0
while (attempts != 0):
    loginPassword = input (">Enter your Password: ")
    # if password matches, login successful and break out of the loop
    if loginPassData == loginPassword:
        print("Login Successful! ")
        break
    # if password doesn't match, login failed, decrement attempts
    else:
        print("Login Not Successful! ")
        attempts = attempts - 1
        # if attempts are 0, notify that user ran out of attempts
        if attempts == 0:
            print("You have run out of attempts! Wait for 2 minutes! ")
            # using time library, sleep for 120s and after that restore attempts to 3
            time.sleep(120)
            attempts = 3
```

This is the last part of the code running:

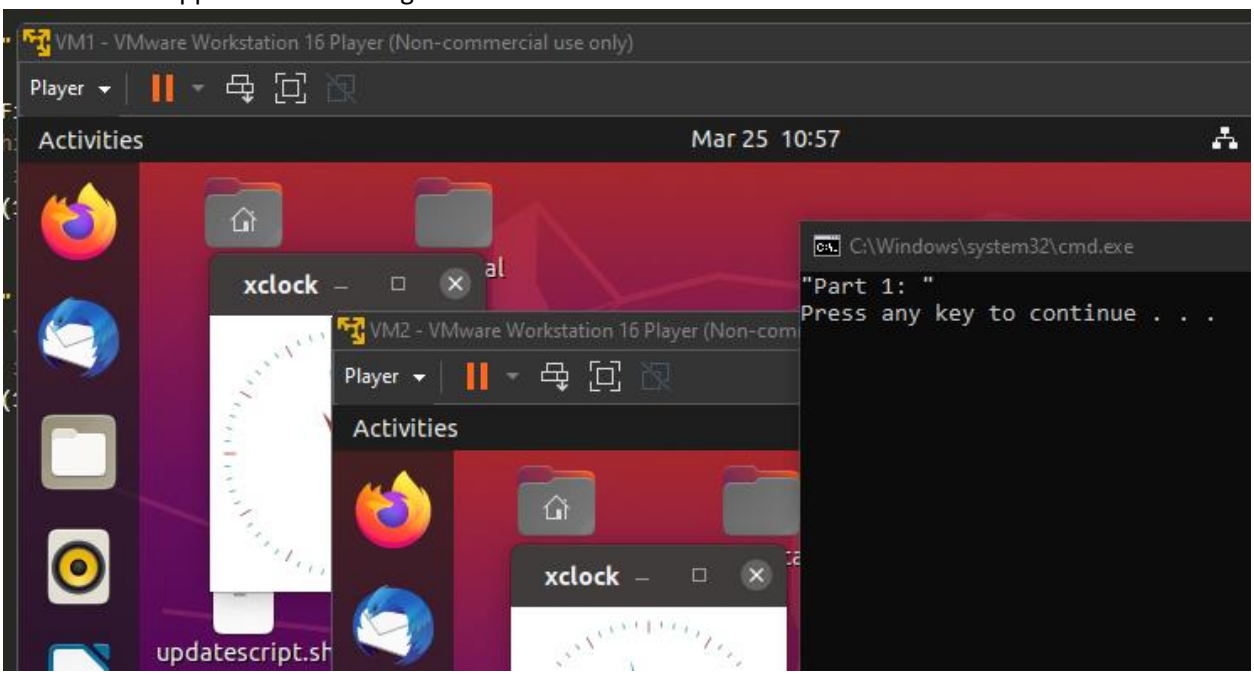
```
>Enter your Login: saim123
>Enter your Password: asdf
Login Not Successful!
>Enter your Password: asdf123
Login Not Successful!
>Enter your Password: 123asfasdaas
Login Not Successful!
You have run out of attempts! Wait for 2 minutes!
>Enter your Password: asdasdasd
Login Not Successful!
>Enter your Password: 43@1Saim
Login Successful!
PS I:\Year 2\Semester 4\Automation\Scripts\CA1>
```

Question 3:

Part 1: First, I make sure that only the output is visible using the **"@echo off**" and "**cls**" to clear the window. It is a simple program that only runs a simple **xclock** and starts it at 0 on 2 virtual machines.
I switch to the **C:** directory and direct myself to the **VMware Player** or **VMware Workstation** folder so I can use the **vmrun** commands. I then use a **for loop** which starts at 1, is incremented by 1 and finishes at 2 which uses a command to **run a program in the guest machine** providing it with all the **arguments** like the path to the **virtual machine**.

```
:: make output visible only and clear screen right after
@echo off
cls
echo "Part 1: "
C:
CD C:\Program Files (x86)\VMware\VMware Player
:: virtual machines must be on before continuing
:: start at 1, increment by 1, end at 2 and do command to run programs in guest machine to display xclock
FOR /L %%I IN (1,1,2) DO vmrun -T ws -gu eryk -gp 1234 runProgramInGuest "I:\VMs\VM%%I\VM%%I.vmx" -noWait /usr/bin/xclock -display :0
pause
```

This is what happens after running the code:



Part 2: Since I have already switched to the **VMware Player** or **VMware Workstation** folder, I don't have to do it again. This time I make another **for loop** the same as before but this time I use the **run script in guest** version of the command. **The syntax of the command is correct in my code.**

```
echo "Part 2: "
:: I have made the updatescript.sh which can be pasted into the guest machine to run it however it does not run due to a bug in gnome or vmware
:: start at 1, increment by 1, end at 2 and do command to run program/script in guest machine to check for updates and update
FOR /L %%I IN (1,1,2) DO vmrun -T ws -gu eryk -gp 1234 runScriptInGuest "I:\VMs\VM%%I\VM%%I.vmx" -noWait /bin/bash /home/eryk/Desktop/updatescript.sh
pause
```

This is the code of the **updatescript.sh** that I am trying to run:

```

echo "1234" | sudo -S apt-get update && sudo -S apt-get upgrade

echo "Successfully Updated! "
```

However due to a bug with gnome, the script does not run through the vmrun command. This is what happens when I run the update script manually **just to prove that it works**: