# Task Three: Scripting

As part of your role, you must train new IT staff automate tasks using scripting. Create a user guide with specific examples to help new staff members understand how scripting can be used to make tasks more efficient. Include at least 5 of the following tasks:

1. Find all scripting commands that are *print* related. Output the list to a local file.
2. Append (the long version of) the current date and time to the same file.
3. Close all notepad files.
4. Append the last 20 errors from the event log to the same file.
5. Append a list of all available WMI classes to the same file.
6. List the start command or full path of every executable (hint: use wmi).
7. Identify what account the spooler is running as (hint: use wmi)

# Scripting Guide for New Users

By Jamie Irvine 50103233

Scripting can be used as part of your day-to-day work to make tasks more efficient and free up your time to handle more complex tasks.

To start you will need to download and install Git Bash from git-scm.com. During installation, enable the option to add Git Bash to the **PATH**.

Next, create a new folder on your computer. Once you have created a folder, right click inside the empty folder and select ‘Open Git bash here’

Create a new file name according to the examples below eg ‘print\_scripts.sh’. This can be done by typing ‘touch print\_scripts.sh’ into the terminal then pressing Enter.

Now that you have your .sh file you can edit it inside an IDE such as PyCharm or Notepad++.

Enter the relevant codes below to your .sh file and save.

Make your .sh file executable by going back into terminal and typing ‘chmod +x print\_scripts.sh’

\*\*print\_scripts.sh only an example, amend this to your relevant file name

To execute your script, type into the terminal ./ then the file name of your .sh file eg ‘./print\_scripts.sh’, or double click on your .sh file from Windows File Explorer

Below are 5 examples of when scripting can be used and how to use it.

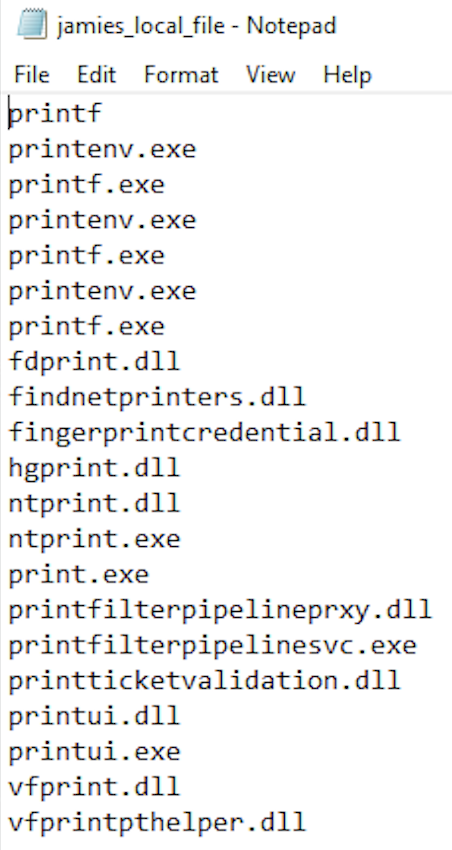
1. **Find all scripting commands that are *print* related. Output the list to a local file.**

To find all scripting commands that are print related and output them to a local file, type the below into your .sh file you created by using an IDE such a Pycharm or Notepad++ to edit the file. Note, you can change the directory and .txt file name in line 7 to suit your preferences.

**A grey screen with orange text

Description automatically generated**

Expected output should be as below, a .txt file will be created in your listed directory and will detail the print related commands on your computer:



1. **Append (the long version of) the current date and time to the same file.**

To append the long version of the current date and time to the same file you can create a new .sh file and type the following code into it by using your IDE of choice. Make sure to use the same directory and file name on line 7 from your previous script.

If you wish, you can include this code in your .sh file from part 1 but be aware that as you add more lines of code to your script it can become complex and error-prone.

A screen shot of a computer

Description automatically generated

Expected output should be the addition of the long date and time to the bottom of your .txt file from part 1.

A screen shot of a computer program

Description automatically generated

1. **Close all notepad files.**

To make a script that closes all notepad files on your computer you can create a new .sh file and type the below code into it by using your preferred IDE to edit the .sh file.

A black background with white text

Description automatically generated

Output should be that any open notepad files will automatically close, see screendumps below:

A screenshot of a computer

Description automatically generated

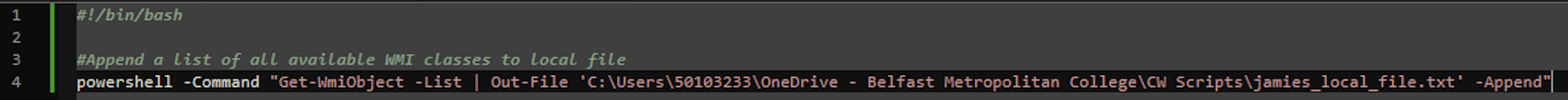
A screenshot of a computer

Description automatically generated

1. **Append a list of all available WMI classes to the same file.**

To append a list of all available WMI classes to the same file you can create a new .sh file and type the following code into it by using your IDE of choice. Make sure to use the same directory and file name on line 4 from your previous script in part 1/part 2.

If you wish, you can include this code in your .sh file from part 1 but be aware that as you add more lines of code to your script it can become complex and error-prone.



Expected output should be the addition of a list of all available WMI classes to the bottom of your .txt file from part 1. As this will generally be a large output, I have only included a small sample in my below screendumps:

A screenshot of a computer

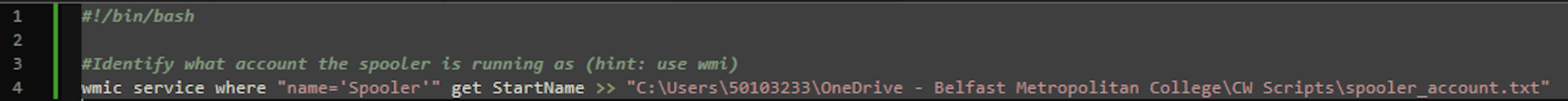
Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. **Identify what account the spooler is running as (hint: use wmi)**

To identify which account the spooler is running as you can type the below code into a new .sh file by using your preferred IDE to edit the file. You can change the directory and .txt file name in line 4 to suit your preferences.



Expected output should be as below, a .txt file detailing which account the spooler is running as:

A screenshot of a computer

Description automatically generated