**Practical (Week 6)**

**A Brief Introduction to the Windows Command Line**

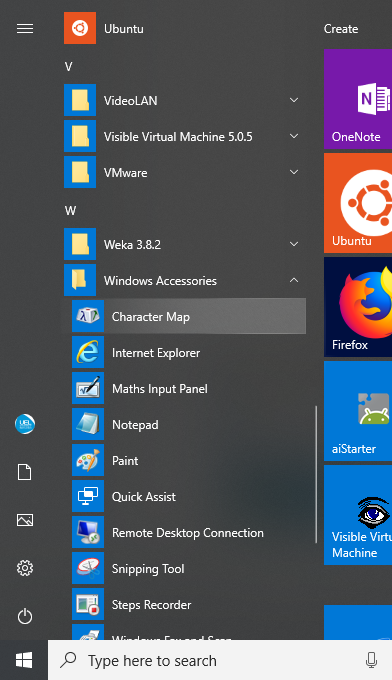
In this practical, you will explore the Windows command line interface. Whilst all of you have used the graphical user interface (GUI) provided by Windows 10, many of you will not have used or even be aware of the facilities that Windows 10 provides its users for interacting with the system via the command line (CLI). The command line may seem strange and difficult to understand at first but it provides a range of powerful features which are used everyday by advanced users such as software engineers, system administrators and cyber security analysts.

**Accessing the Windows Command Line**

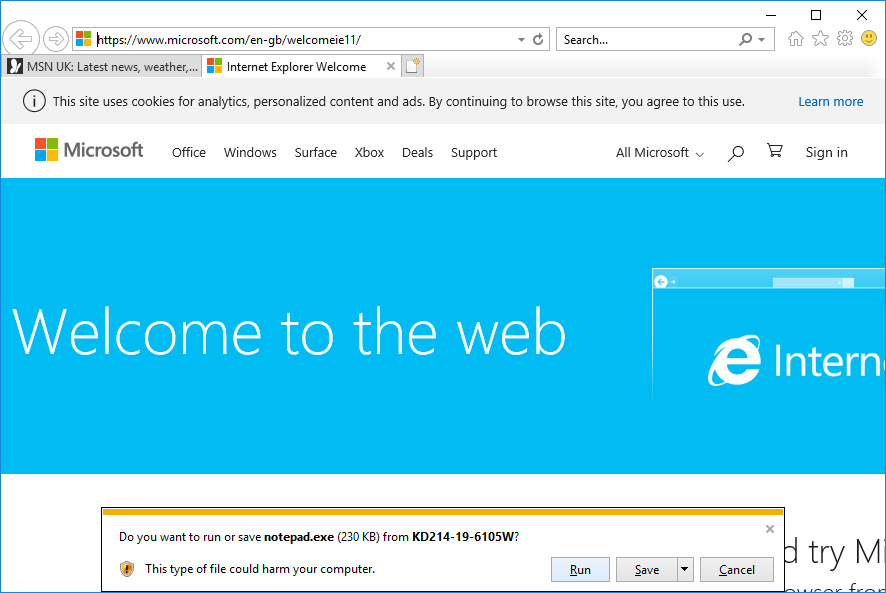
There are several ways to access the command line in Windows 10. Try the following. In each case, make a mental note of what happens.

1. You can use the address bar in Internet (or File) Explorer, which you can find in Windows Accessories.

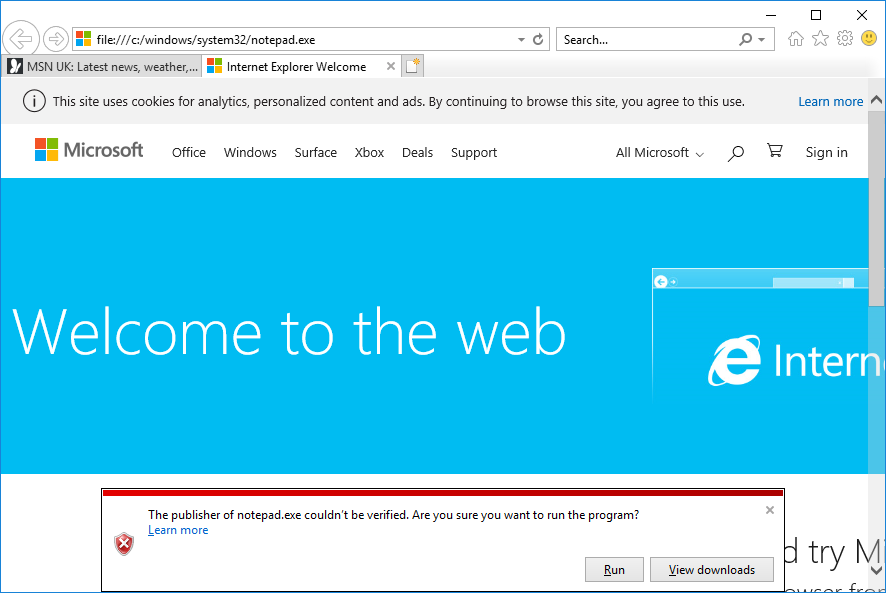
Click on the Windows icon at the bottom left corner, scroll to Windows Accessories and click on Internet Explorer: You will see the following displays.



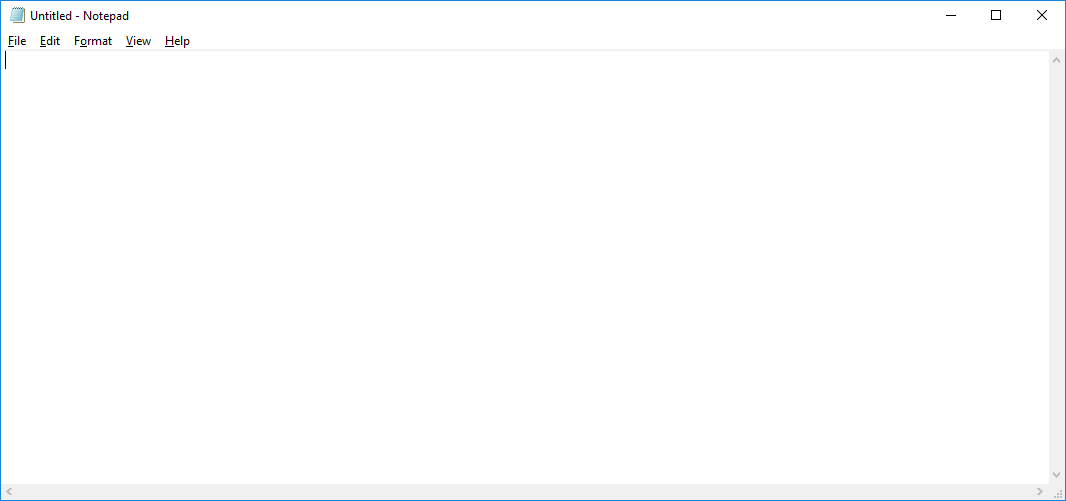
Enter: c:\windows\system32\notepad.exe



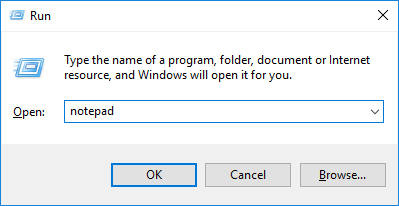
When prompted by this window, click on Run.



Click Run again.



1. Press the Windows key (between Ctrl and Alt keys) and R [Win + R]. When prompted with the following window, type notepad.



1. Go back to the Windows main menu (Click on the Windows icon), scroll to Windows System. Click on it. Which applications are displayed under Windows System?

Answer:

1. Click on Command Prompt.
2. Go back to Windows System.
3. Click on Run and type cmd.
4. Type cmd in the search box in the Windows bar (bottom) and press Enter.

What is the difference between actions a, c and d?



If all went well, the command prompt should have appeared on the screen in all three cases. Now repeat the above steps but this time use the *netstat* command. What happens this time? In the case of steps c. and d. the output from the *netstat* command appears in a window momentarily but not long enough for you to view its contents. In the case of step a., the Command Prompt window remains open after the *netstat* command has finished. *Netstat* (a very useful command for network analysis) is an example of what Windows calls a console application. Unlike *notepad*, which requires the Windows desktop GUI to run, console applications need to be run using the command prompt.

What does netstat do?

From now on, you will need to use the command prompt.

**Windows Command Prompt Basics**

Enter *cmd* in the run dialog on the start menu.

**Getting Help with Windows Commands**

You can get help for a Windows command by typing the *commandname /?*, e.g. *dir /?*

Enter *set /?* to find out about the set command. Make some brief notes.

|  |
| --- |
|  |

**Useful Windows Commands**

The following list contains a number of useful Windows commands. Try each one of them, making a note below of what they do. (Remember, if you aren’t sure, use the Windows help system.)

1. *cd*
2. *cd ..*
3. *cd \windows\system32*
4. *cls*
5. *dir*
6. *dir /q*
7. *dir /ad*

**Redirection**

The Windows command line interface will also let you redirect input and output. Normally the command prompt writes the output from a program to the screen (referred to as the standard output) and reads data into commands from the keyboard (referred to as the standard input). Using redirection, we can alter this behaviour. The following exercises will clarify these points.

Before 1, enter *exit* to close the current command prompt and open a new one so that you are back in your home directory.

1. *echo a line of text*
2. *echo a line of text > textfile*
3. *type textfile*
4. *copy textfile newtextfile*
5. *ren newtextfile oldtextfile*
6. *del textfile oldtextfile*

After 4, 5 and 6, enter *dir* to check what changes have taken place to the files in your directory (folder). If there are a large number of files in the directory, enter *dir /p* to view them one screen at a time.

Make some notes below.

|  |
| --- |
|  |

**Pipes**

Pipes are used to redirect the standard output from one command into the standard input of a second command.

The Windows command line interface supports this feature. To illustrate this, open a new command prompt and enter the following commands, making a note below of what they do:

1. *dir > mylist.txt* (followed by *more mylist.txt* to display the contents of mylist.txt. Pressing the space bar causes the *more* command to scroll text a whole screen at a time.)
2. *dir c:\windows >> mylist.txt* (followed again by *more mylist.txt*)
3. *sort /+12 < mylist.txt*
4. *dir c:\windows | more*
5. *type mylist.txt | sort* (Note that in this case *type* is an actual command. What is the difference between this combination of commands and those in c. above?)
6. *dir c:\windows | find “log” /i | more*

|  |
| --- |
|  |

**Using Wildcards**

The Windows command line support wildcards. Wildcards can be used to manipulate multiple files with a single command. Windows uses two wildcards, \* and ?. ‘\*’ means zero or more characters whilst ‘?’ means a single character.

Enter the following commands which illustrate the use of wildcards. Make notes below of the differences between the various commands.

i) *dir c:\windows\\*log\** (How is this different from f. above?)

ii) *dir c:\windows\\*.log* (How is this different from i?)

iii) *dir c:\windows\w\**

iv) *dir c:\windows\?i\**

v) *dir c:\windows\???.\**

vi) *dir c:\windows\????.\** (How is this different from v?)

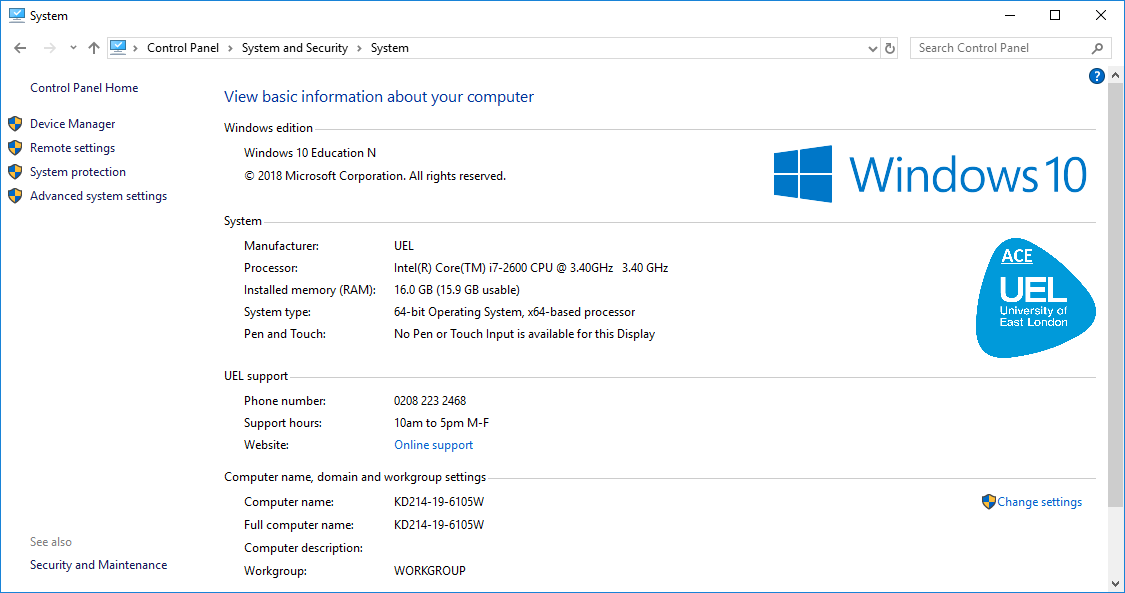
|  |
| --- |
|  |

Wildcards are extremely useful. As well as using them directly at the command prompt, you can use them within batch files and scripts.

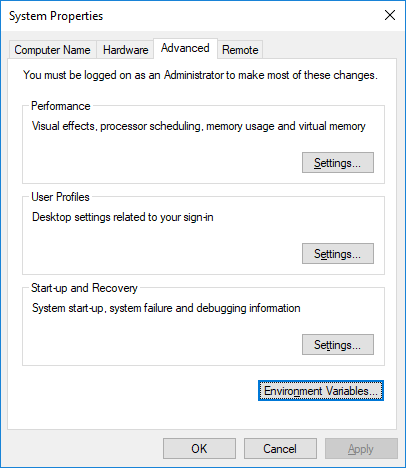
**Environment Variables**

Windows makes use of environment variables. As the name suggests, an environment variable is used to store information about the environment in which a process executes. (Remember that each time we execute a program, a process is created. That process can obtain information about its environment by querying the environment variables.) You can inspect environment variables in Windows by going through

Windows System → Control Panel → System



Click on Advanced System Settings.



Click on Advanced → Environment variables, making notes of what you find.

|  |
| --- |
|  |

From the command prompt, enter *set* to view the same information.

You can display the value of an individual environment variable using either *set variable* or *echo %variable%*.

Enter the following commands

* *set os*
* *echo %os%*

You should get the same answer! Using either *set* or *echo*, find the values of the following environment variables; *computername*, *comspec*, *homepath*, *path*, *temp* and *username*. What purpose do each of these environment variables serve?

|  |
| --- |
|  |

You can set environment variables using the *set* command.

Enter *set workdir=c:\windows* then enter *echo %workdir%* to check that the variable has been correctly set.

You can also use environment variables within commands.

Enter *cd %workdir%*. Are you in the right directory?

**Command History**

You can scroll through previous commands by pressing the ↑ and ↓ keys.

And finally

enter *exit* to close the command prompt.

In your own time, you should search the Web (www.ss64.com is a good starting point) for Windows command prompt tutorials that will help you to develop your command line skills further.