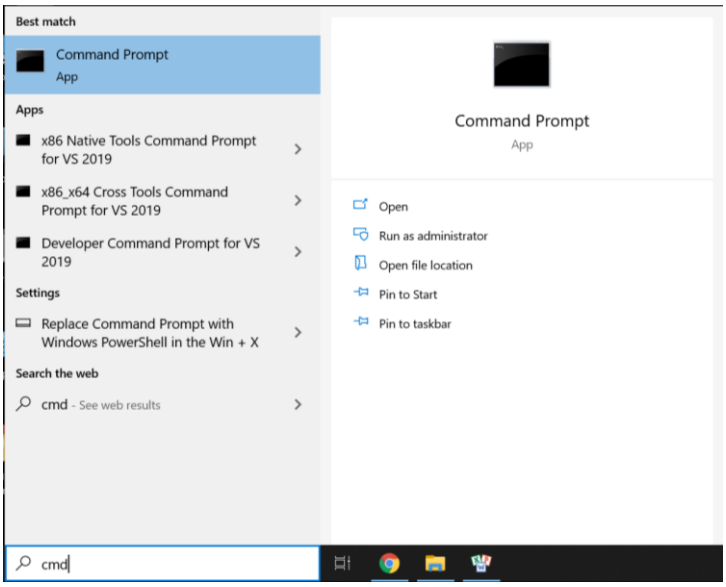
UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

Practical 1 : Command Line Interface (CLI)

In this lab, you will use Windows CLI commands to manage folders, files and programs. Thus, you need to have a computer running Windows as the main Operating System.

Navigate to “Run” or go to windows search section and typed in “cmd”.



The command line interface (CLI) windows is opened as below.

Figure 1: CLI

Based on your CLI, which directory / path are you at?
For Figure 1, it’s simply Yim Ling / C:\Users\Yim Ling

Now, change the directory/path to root directory / C:\
Just need to type “cd ..” then press “Enter”.
Repeat it and you’ll find yourself redirected to the root directory as shown in Figure 2.

Figure 2: redirect to root directory.

CD or change (your working) directory simply means you can navigate between directories within a path.
Another way of changing directory straight to root directory is simply typing "cd/", then press "Enter".

In order to know the whole list executions you can do with CLI, you may type in "help" then press "Enter"

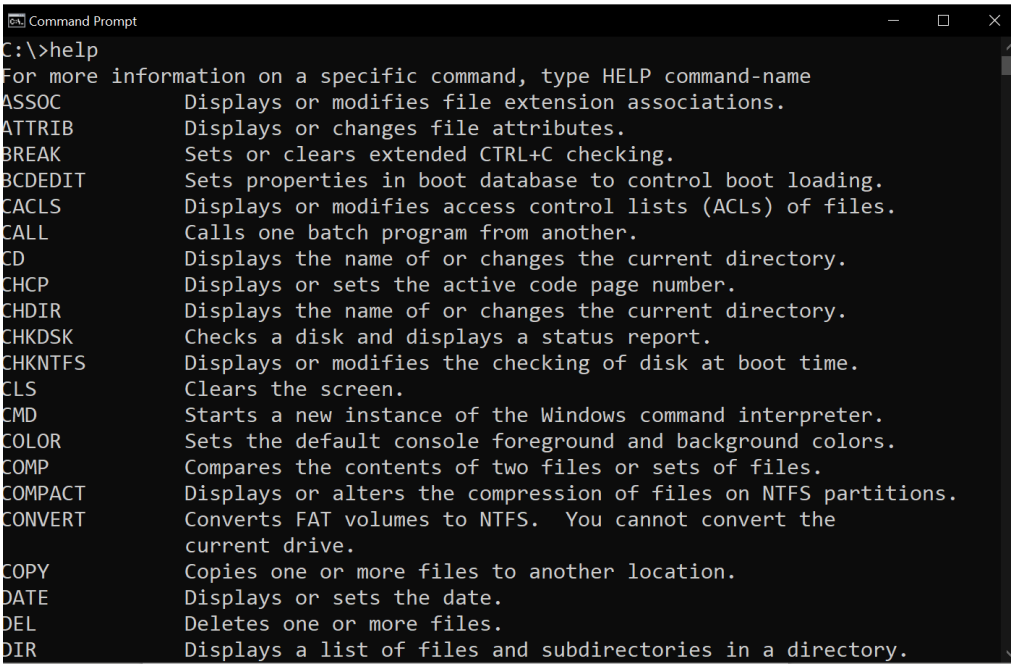


Figure 3: List of commands in CLI.

Now let's visit the basic commands of exploring different directories and files in a Windows system.

While in the root directory, type "dir" then press "Enter"
While looking at the listed Directory of C:\ in the CLI, navigate using Windows GUI to C:\ Directory. What could be derived between these two opened windows?

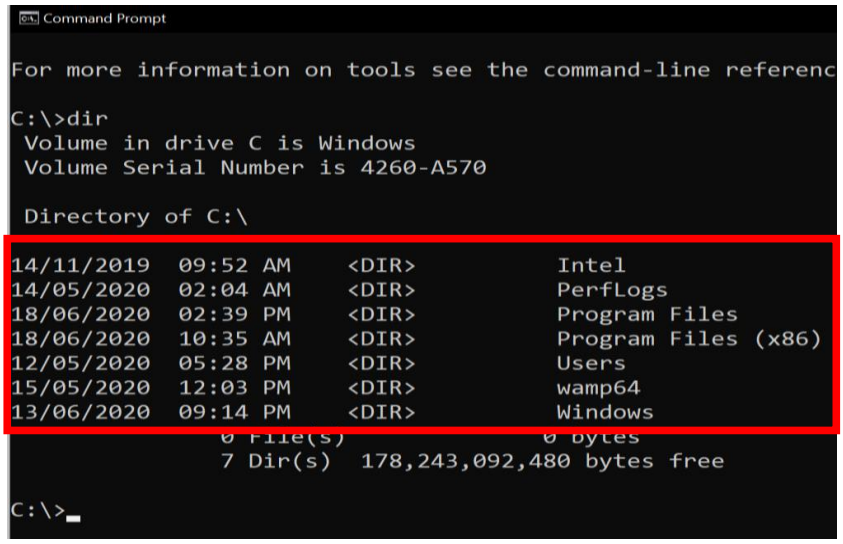


Figure 4: Details of directory C: in CLI

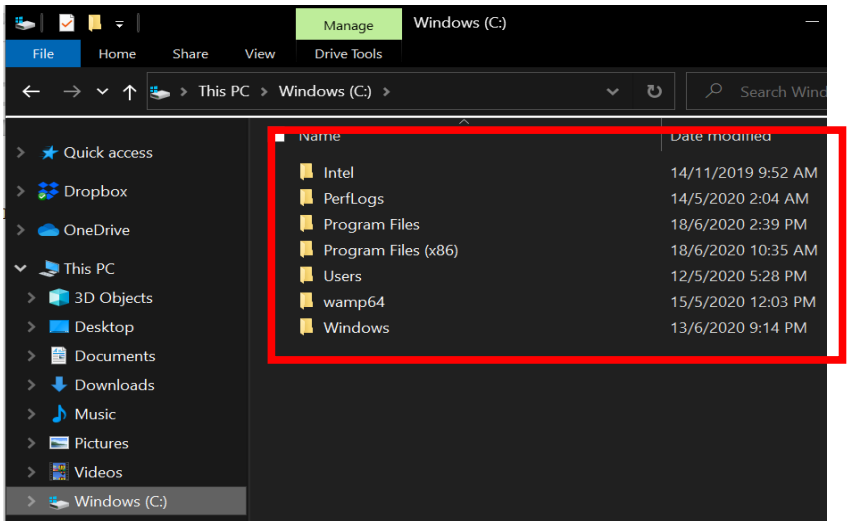


Figure 5: Details of directory C: in GUI

Now, using Windows CLI, let's navigate to Desktop\ directory and use the basic commands to create new directories and files.

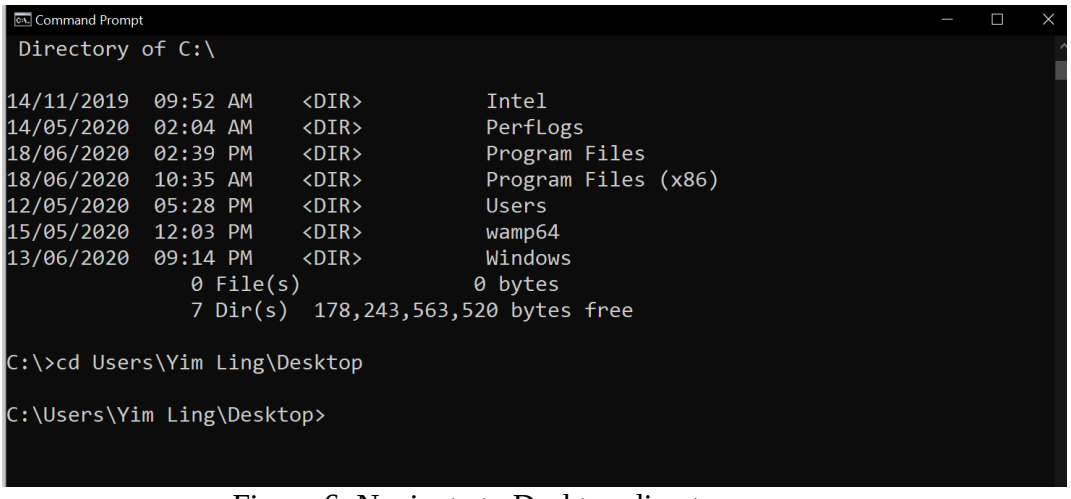


Figure 6: Navigate to Desktop directory.

Create a new directory using "md" and "mkdir" command.

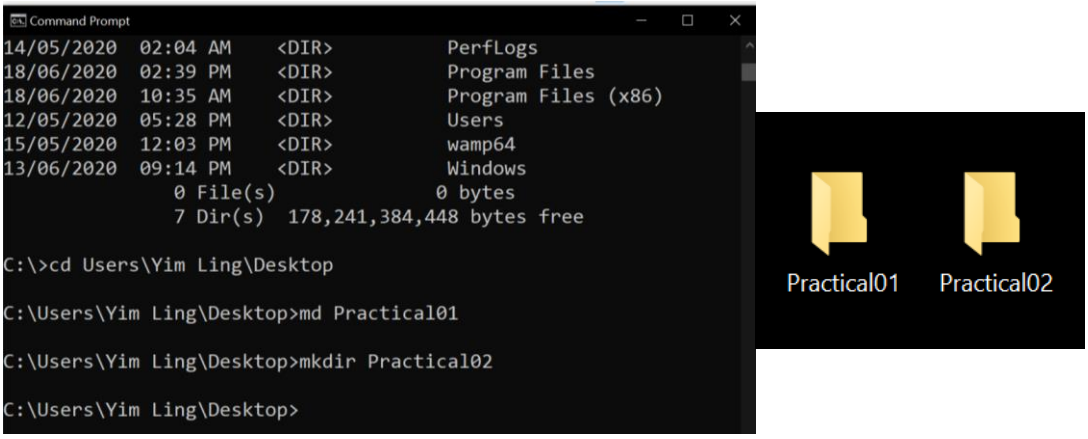


Figure 7: Creation of Practical01 and Practical02 in Desktop directory

Create an empty textfile in Practical01 using "type" command in CLI.

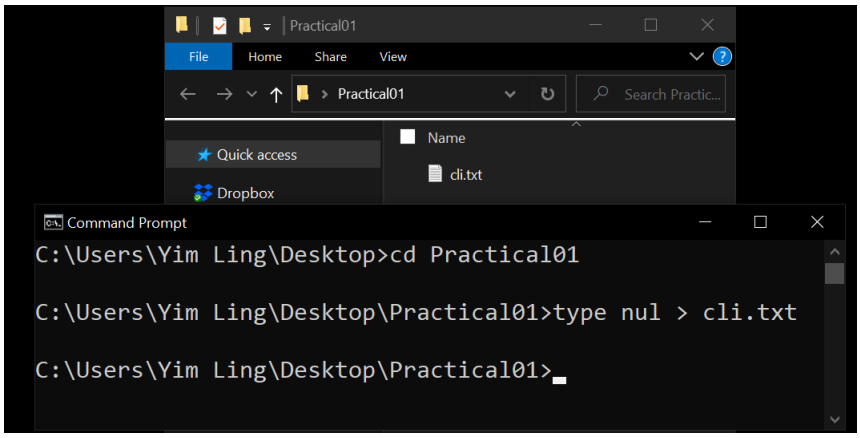


Figure 8: Creation of empty textfile in Practical01 directory.

Now, add some texts into the empty texfile using "echo" command.

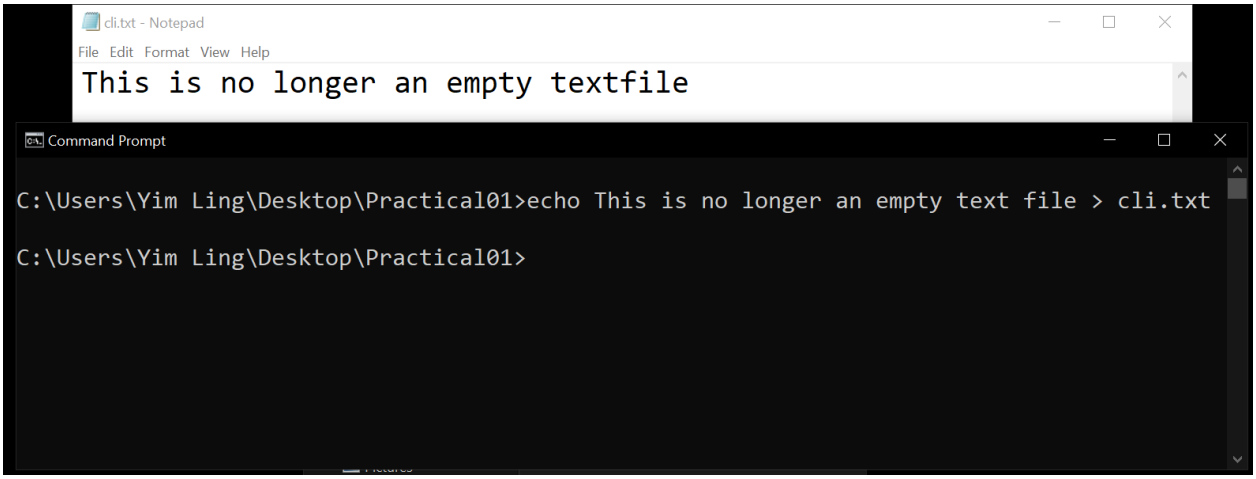


Figure 9: Add texts into an empty text file.

Exercises:

Make changes to the directories where eventually Figure 10 is achieved.

```
C:.
|   5.3.7.2 Lab - Working with CLI Commands in Windows.pdf
|   Microsoft Edge.lnk
|   Microsoft Teams.lnk
|   Start Zoom.lnk
|
+---Practical01
|       cli.html
|       cli.txt
|
\---Practical02
        bronze sculptures 01.jpg
        cli.docx
        KA block 01.jpg
```

Figure 10: Graphical view of desktop directory.

Detailed requirements;

1) An additional file “cli.html” shall be created in Practical01 directory. Using CLI command evoke an editor to edit “cli.html” so that the file generate webpage as shown in Figure 11.

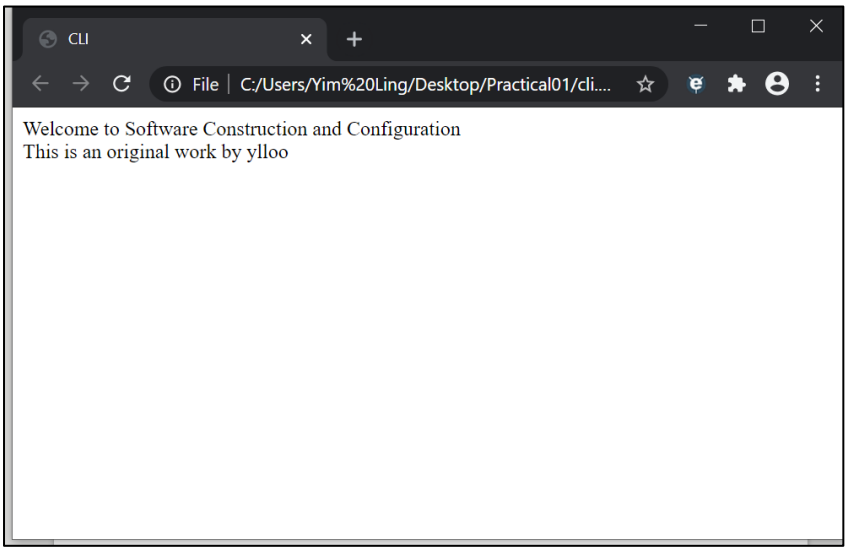


Figure 11: cli.html

2) Create an empty .docx document file in Practical02 directory.
3) Copy all files from Pictures folder (download some pictures first) into Practical02 directory.
4) Using CLI command, generate a graphical view of your desktop directory; similar to Figure 10.

Things to remember:
1. The first thing you type into the CLI is actually just the name of a command.
2. The things that come after the command being called are called "arguments", and they are passed to the command being called.
3. CLI is very sensitive to spaces.

Commands with Appropriate Arguments.

Using a combination of [command] [/?] in CLI, you get to know what are the arguments available for the command.

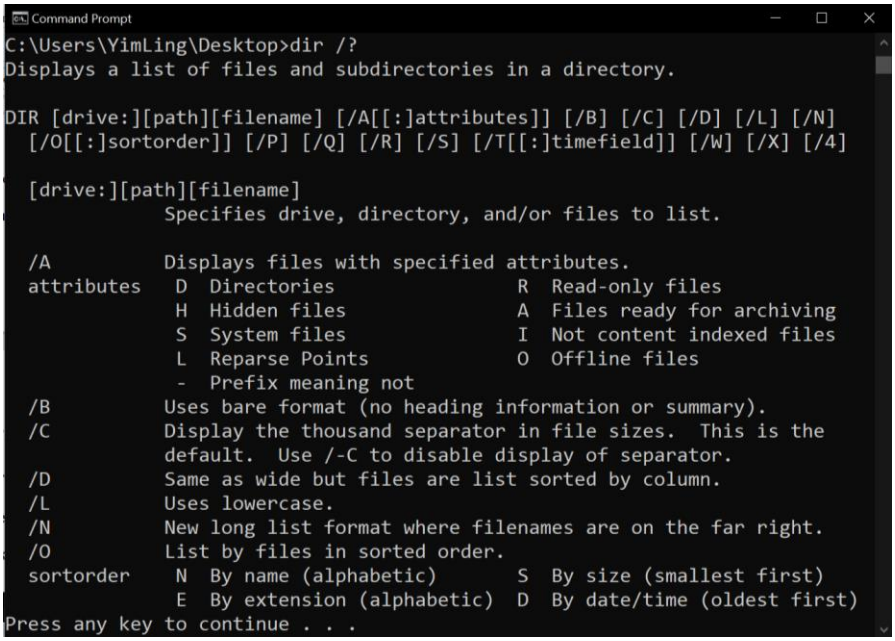


Figure 12: Displaying lists of arguments for a function.

1. Displaying the directory with specific attributes will need to type "dir /a:[attribute]" then press "Enter"

Figure 13: An example of displaying files with specific attribute.

Q: What was the command and argument used in Figure 13? What was implicated?

2. Displaying the directory with specific sorting mechanism will need to type "dir /o:[attribute]" then press "Enter"

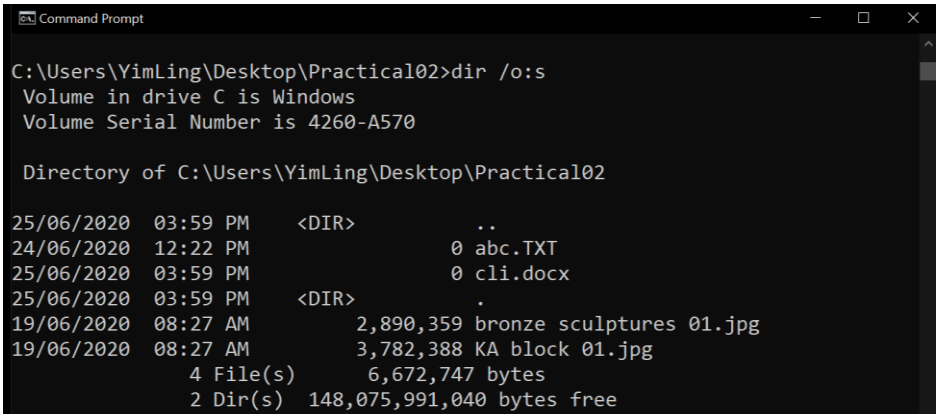


Figure 14: An example of displaying files with specific sorting mechanism.

Q: What was the command and argument used in Figure 14? What was implicated?

3. Displaying a directory may contain multiple arguments for display customization.

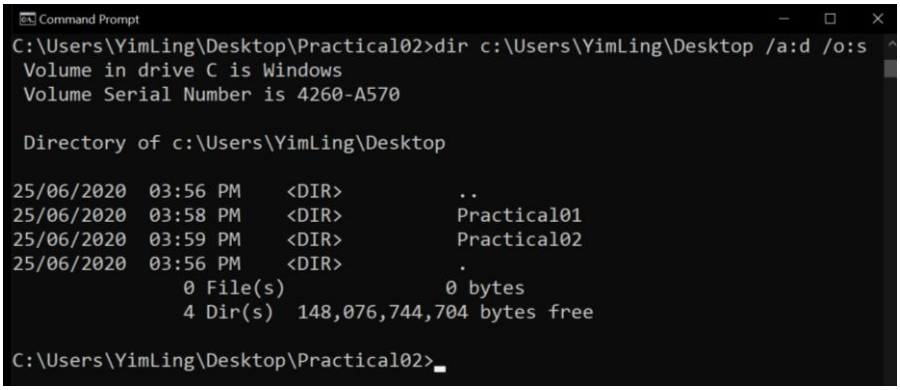


Figure 15: Displaying a directory with combination of arguments.

Q: What was the command and arguments used in Figure 15? What was implicated?

Wildcards.

1. The usage of wildcards are inevitable especially in command line programs.

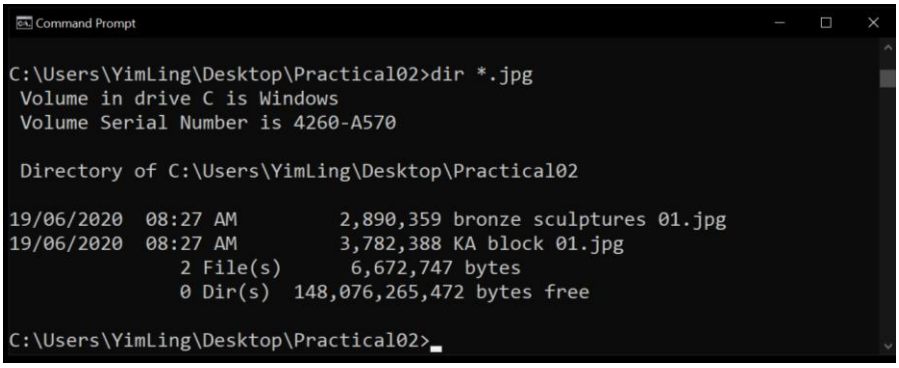


Figure 16: Use of wildcards to display files 1.

Q: What was the command and argument used in Figure 16? What was implicated?

2. Create an empty textfile in Practical01 using “type” command in CLI.

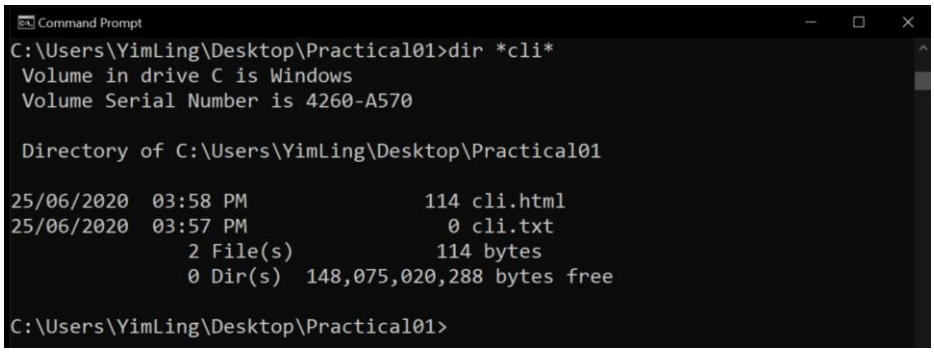


Figure 17: Use of wildcards to display files 2.

Q: What was the command and argument used in Figure 17? What was implicated?

Piping.

1. Echo the “dir” command

   Before we explore piping in CLI, let us explore an additional use of “dir” command. Rather than listing the current working directory in CLI, “dir” is able to save the listing in a specified text file. Type “dir > cli2.txt” and press “Enter”.

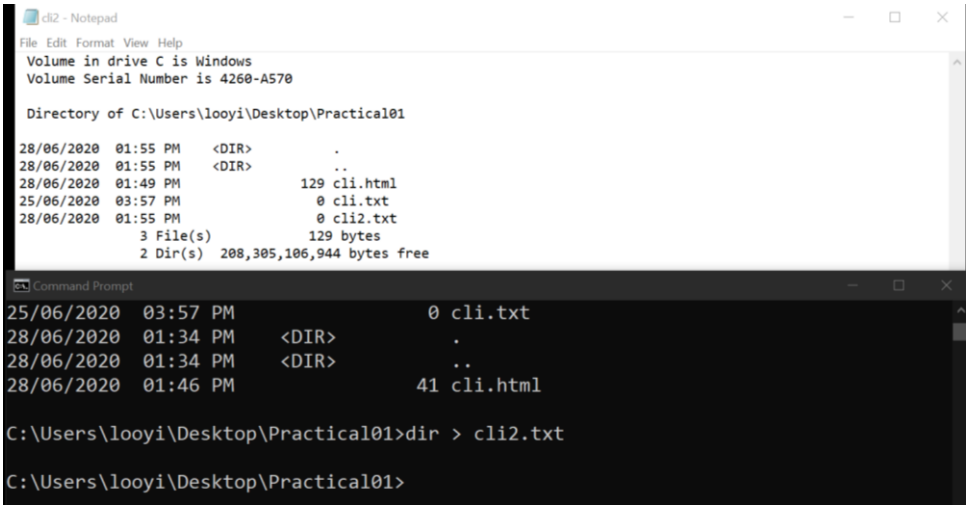


Figure 18: Echo directory details into a file.

Q: What was the command used in Figure 18? What was implicated?

2. The “sort” command.

   Now, explore the “sort” command in CLI. Type “sort < cli.txt” and press “Enter”. Press “Enter” after each line. When you are done, press “Ctrl+Z” then press “Enter” again. Ctrl+Z is a special key that tells the CLI to stop sending keyboard input to the program.

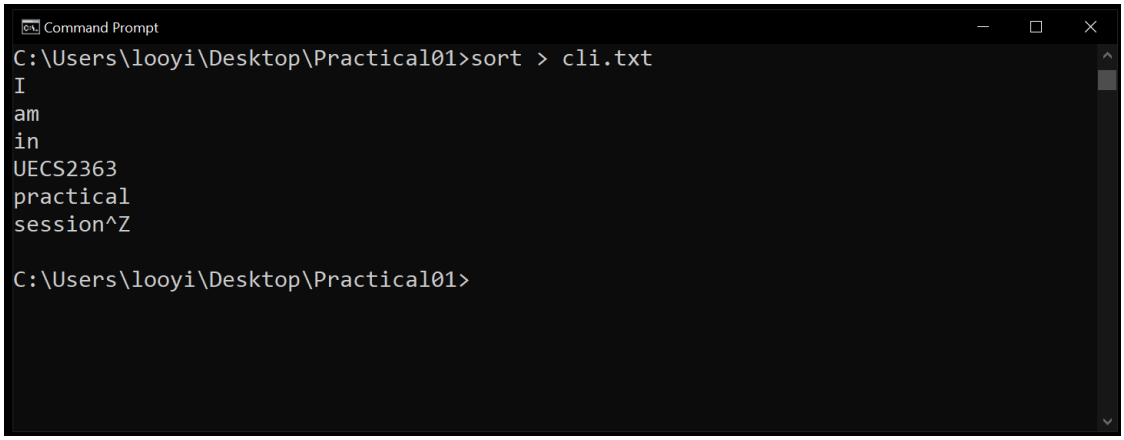


Figure 19: Using sort command.

Q: What was implicated after executing similar steps in Figure 19?

3. Use of piping in CLI.
   Commands "dir" listed out details of the current directory while "sort" is used for sorting (usually ascending). What if combining both commands in a command line? This will be called piping. Pipe or "|" on the keyboard, allows combination of both commands be done in CLI.

   Type "dir > cli3.txt" and press "Enter" to echo the details of the current working directory into "cli3.txt" as shown in Figure 20.

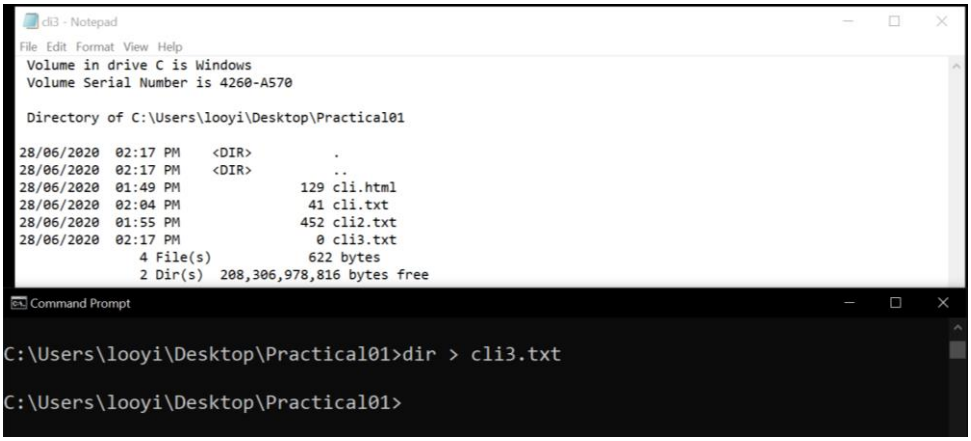


Figure 20: Echo directory details into cli3.txt.

Now, type "dir | sort" and compare the output in CLI with "cli3.txt" as shown in Figure 21.

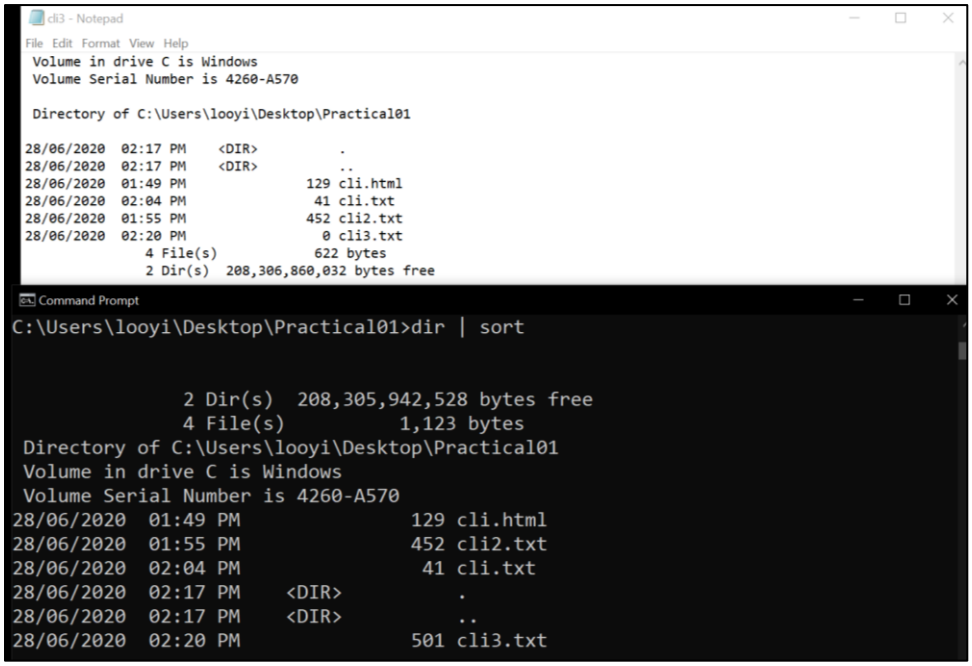


Figure 21: Piping in CLI.

Q: What was implicated after execution of the command line as shown in Figure 21?

4. Use of "more" command.
   Another useful command for list of display on CLI is "more" command. Try by typing "dir C:\Windows | more" and press "Enter".

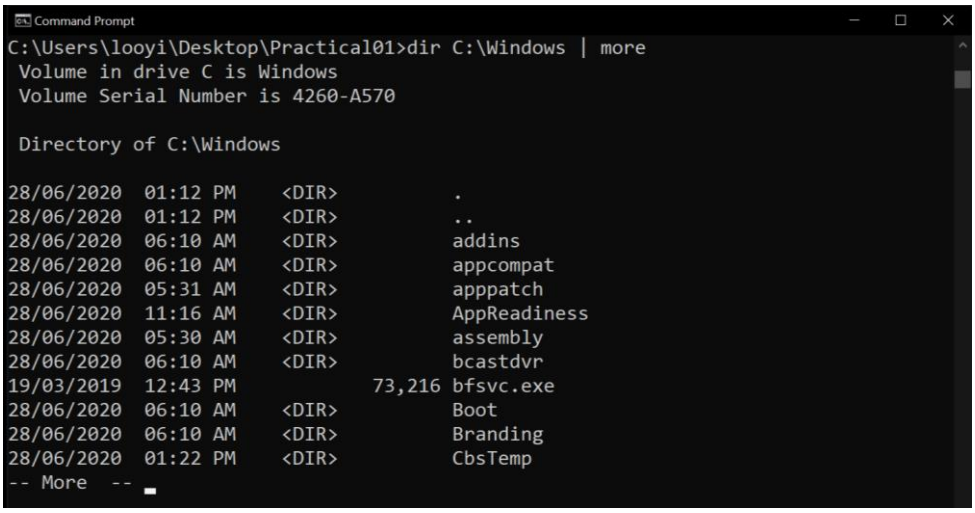


Figure 22: Piping "more" to "dir" in CLI.

Q: What was implicated after execution of the command line as shown in Figure 22?

Batch Files.

Submission of a software project / assignment by email usually use a common way of zipping up all files and send or upload to an email address or a cloud storage. Normally one would use a program such as WinZip for this task. We all know what are the steps; click on the WinZip icon. Click, click,

click, until each file is added. Click, click, click, type the zip file name, click, click, and it is done. If a last-minute change is needed to be made? Do it all over again. And again...

1. Using the "jar" command
   Before we explore batch files, let's explore the basic command, "jar" to create zip file in CLI. In order to use this command a JDK need to be installed in the machine. "jar" command could only work in the folder where "java.exe" is, which normally is in the "C:\Program Files\Java\jdk-14\bin\".
   CD to the directory containing "java.exe" Type "jar -cvf %destinationpath%\cli.zip * %destinationpath%" and press "Enter".
   Note: "-cf" denotes compression to a file, while v option in "-cvf" produces verbose output, which lists all the actions that "jar" command takes.

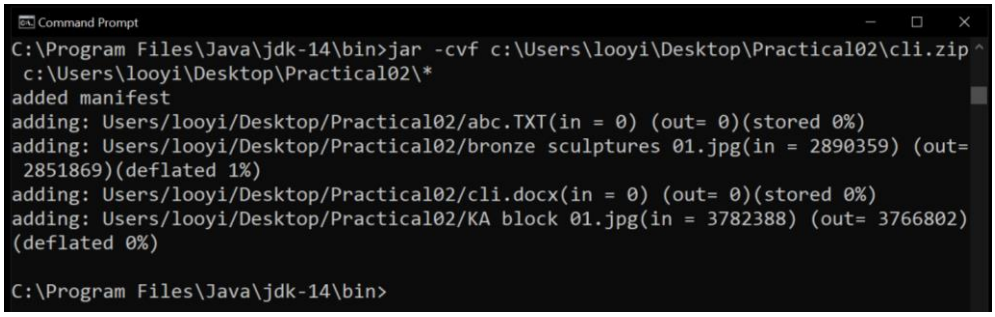


Figure 23: Create a zip file.

Q: What was implicated in Figure 23?

2. Using batch files.
   Rather than typing the command every time, you can place it into a batch file, and run the batch file as a single command. We will start with a simple version of such a batch file.

   Start the Notepad program and save the file as "clizipbatch.bat" with texts entered into the file as shown in Figure 24.

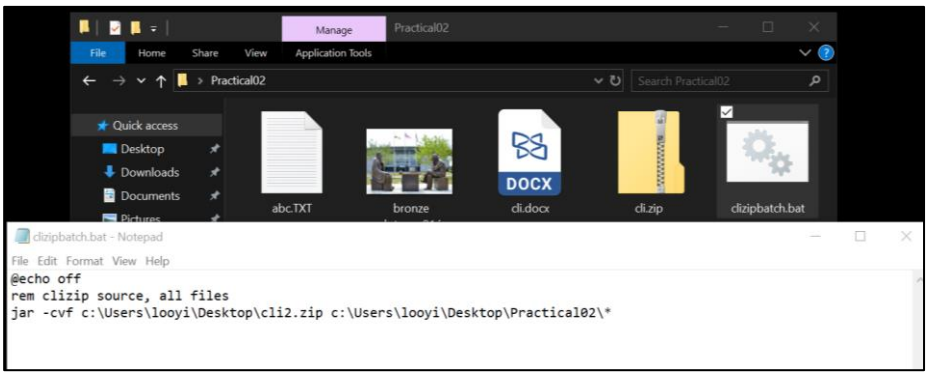


Figure 24: Create a batch file for zip function.

Now launch clizipbatch.bat file in the CLI as shown in Figure 25.

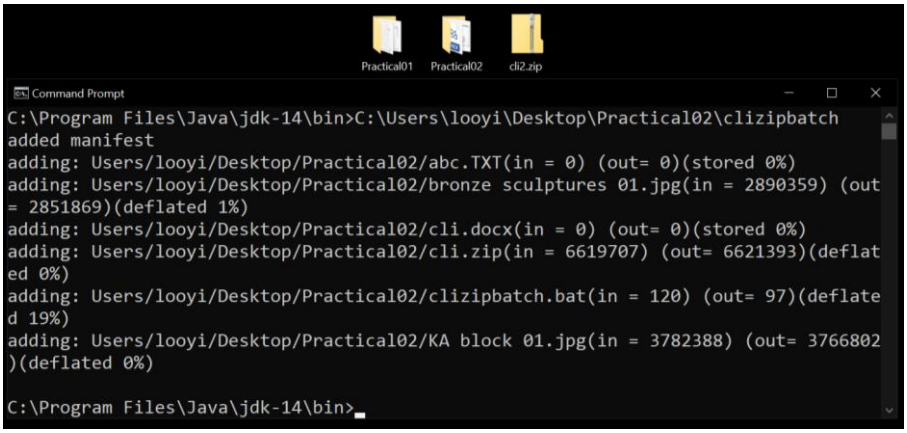


Figure 25: Run clizipbatch.bat in CLI.

Q: What was implicated in Figure 25?

Now edit the clizipbatch.bat file. The first two commands of the start batch file are new commands. "@echo off" suppresses the display of each command on the screen as it is executed. A line starting with "rem" contains comments that are ignored by CLI. Try adding "rem" before "@echo off" in the first line of clizipbatch.bat file. Then execute the file once again in CLI as shown in Figure 26.

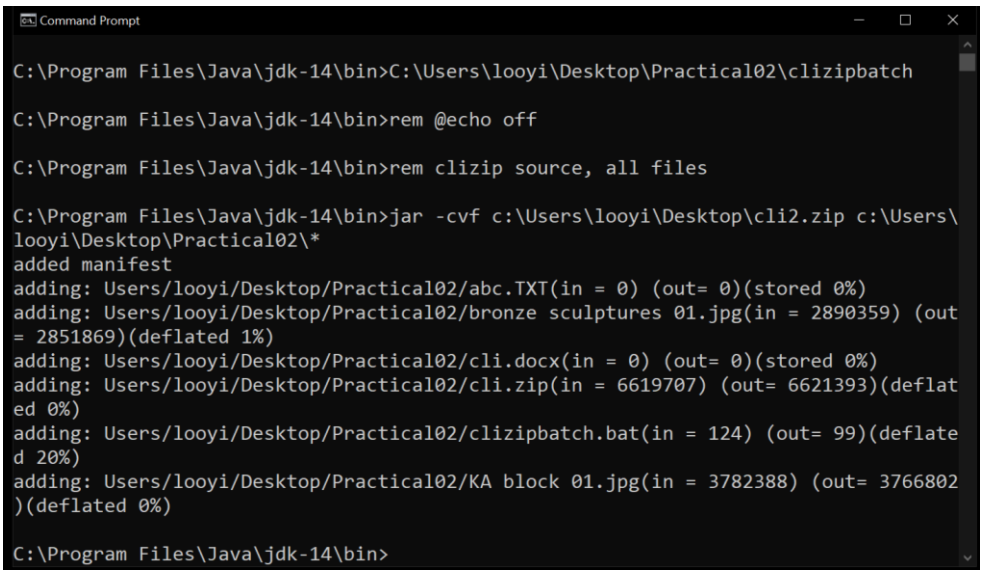


Figure 26: Run edited clizipbatch.bat in CLI.

Q: What was implicated in Figure 26?

Simple Programming.
Simple programming can be done in batch files such as conditional or loop statements.

1. Conditional statements
   In batch files, we may write simple programming for conditional statements to be executed. Create a batch file as shown in Figure 27 in Practical01 folder. Execute the batch file in CLI.

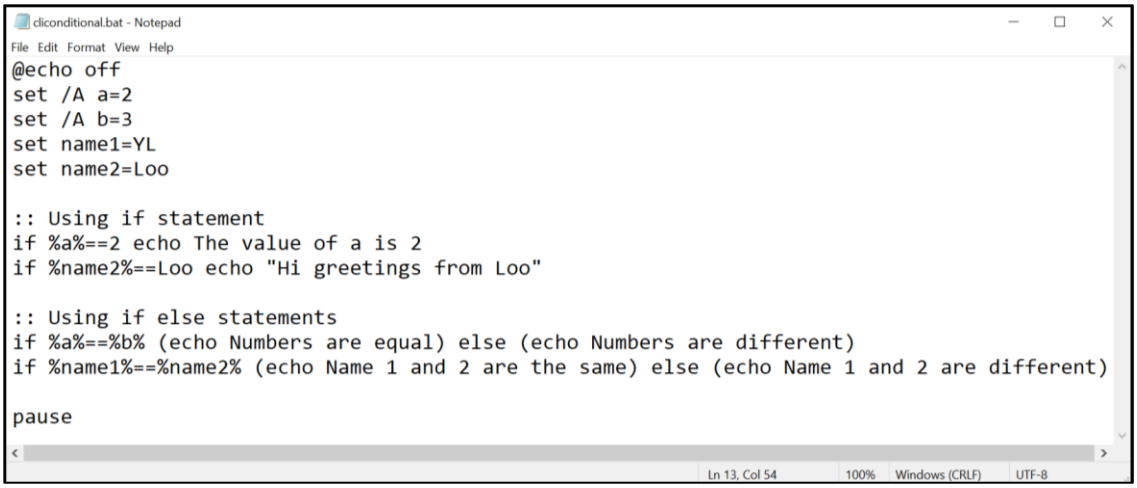


Figure 27: Create a batch file that contains conditional statements.

Q: What was implicated after executing batch file of Figure 27? What will be implicated if "@echo off" is removed?

2. Loop statements
   Now, let's look into having loop or repeating statements in batch files. Create a batch file as shown in Figure 28 in Practical01 folder. Then execute the batch file in CLI.

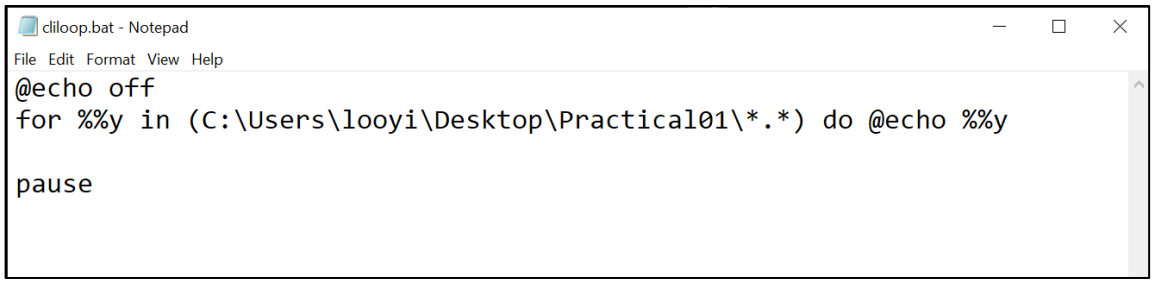


Figure 28: Create a batch file that contains loop/repeating statements.

Q: What was implicated after executing batch file of Figure 28? How to loop through directories instead of files?