

# CIS 350 INFRASTRUCTURE TECHNOLOGIES

## LAB #1

**Due date: See Blackboard**

**Objectives:** Learn about **Windows Command Prompt/Line**

### **Brief Review of MS-DOS**

MS-DOS was the primary operating systems (OS) for IBM PC computers. It was developed by Microsoft in the beginning of 1980s. MS-DOS was a 16-bit OS which dominated the market for about 15 years. It was a very important OS used on single-user, single-program, stand-alone desktop computer systems. MS-DOS advantages are its fundamental operation and its straightforward user commands issued from the prompt. It had two disadvantages, however. The first was its lack of flexibility and limited ability to meet the needs of programmers and experienced users. The second stemmed from its roots; it was written for a single family of microprocessors, the Intel family of chips: 8086, 8088, 80186, and 80286. When microcomputers using those 8- or 16-bit microprocessors dominated the market, MS-DOS did it too. But newer 32- and 64-bit chips have made inroads and MS-DOS had to adapt or make way for other, more sophisticated, systems such as Windows 1.0, 3.1, 3.11, NT, 2000, XP, Vista, 7, 8, and 10. Early versions 1.0 and 3.1 of Windows were very awkward and slow as these were application programs running on top of MS-DOS.

### **The Windows Command Prompt/Line**

There is a common misconception that the command prompt in Windows10 is the same as MS-DOS. There are some superficial resemblances and some commands with the same name, but in fact the old 16-bit MS-DOS is dead. All remnants of MS-DOS are totally gone from the Windows 10 kernel (though there is still a DOS *emulator* for legacy programs). Windows 10 is a 32-bit or a 64-bit protected memory system with a very different approach from the MS-DOS. The command line in Windows 10 has many more capabilities and none of the 16-bit limitations like the restriction to the MS-DOS file name format (8 characters for a filename and 3 characters for a filename extension). The augmented capabilities make the Windows command prompt/line a powerful and very useful tool today.

### **Reasons for Using the Windows Command Prompt/Line**

- (1) GUI systems implemented in Windows, MAC OS, or Unix/Linux hide what is effectively happening inside the computer. Windows command prompt/line is the system that is more basic and direct. Consequently, if you understand what is happening in Windows command line, you may find it easier to understand what is happening behind GUI in Windows, MAC OS, Unix/Linux, or other more complex operating systems.
- (2) Windows command prompt was developed after Unix. Many Windows commands such as redirection, piping, and filtering are very similar to Unix/Linux commands. As a result, you can transfer experience and knowledge acquired in this lab to Unix/Linux.
- (3) A virus may attack your computer. Virus protection programs include a special recovery CD, DVD, or thumb drive that uses Windows command prompt to support the recovery process.
- (4) For professional programmers the command line interface still shines through over the GUI.

## **Topics Covered in the Lab**

Because the lab reviews very basic Windows commands such as *dir*, *copy*, *format*, *erase*, *chkdsk*, *tree*, *mkdir*, *rmdir*, *chkdir*, *sort*, and *type* as well as the input/output redirection and piping concepts. Also, the lab introduces students to the Windows command and upside-down tree directory structure, default/current drive, default/current directory, root directory, path name, file naming conventions, and wild cards. You can transfer much of this experience to the Linux operating system that uses similar or the same concepts.

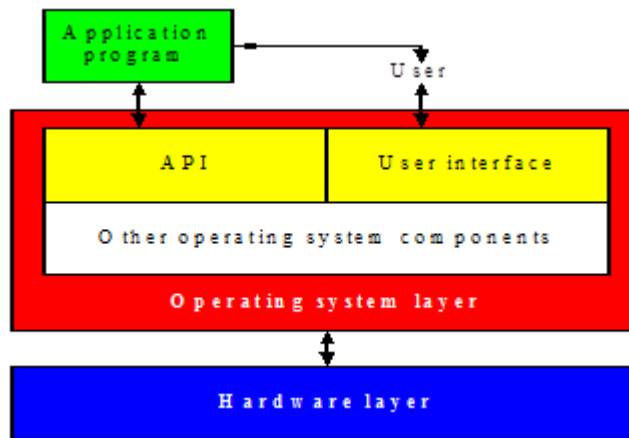
The command set in Windows 10 is obviously much more comprehensive than the commands that you will use/see in this lab. You may search the Internet for more Windows commands. For example, a link <http://ss64.com/nt/> provides a more comprehensive list of Windows commands, their description and syntax. You may search the Internet for a full set of Windows 10 commands, including commands useful for administrators of the Windows system. Due to space constraints, we will not run any of these commands in this lab as Lab 1 already has 25 pages.

## **Important Notes**

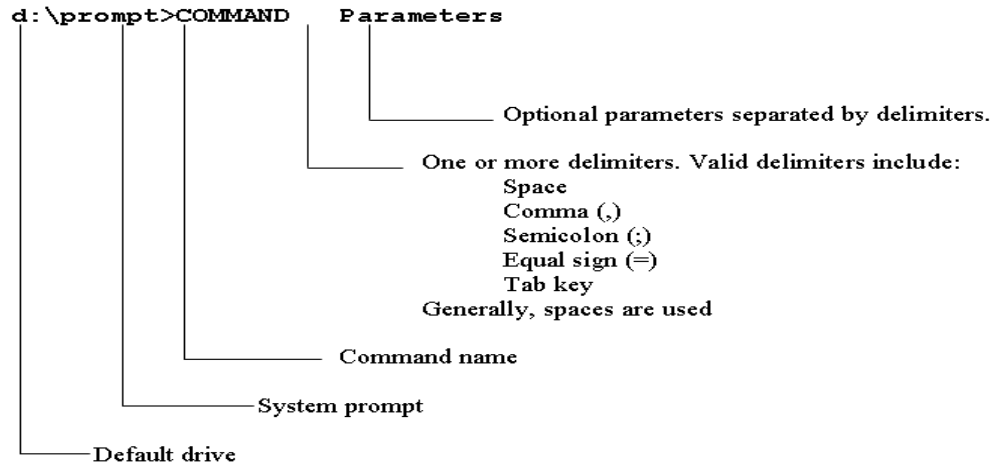
1. You will need a thumb drive, preferably blank. If you have some old work on it, it is OK.
2. In this lab you have to create several text files. You may do it with the Windows Notepad or COPY command. You will also copy some of the files into different directories, which you will create, so that each directory has something in it.
3. If you follow the lab instructions carefully, your thumb drive should have all the required directories and files on it.
4. If you work this lab on a college lab computer using VMware, you may not be able to perform several commands. For example, the *tree* command may not work as it may be disabled by the system administrator. Output redirection to system devices may not work either.
5. Note that many concepts such as input/output redirection and piping covered in this lab will apply to UNIX/Linux covered in Labs 3-5.
6. If you notice any errors in this lab or suggest any improvements, please bring it to my attention in Lab 1 Report. See point 8 below.
7. Note that the steps in this lab were performed on the instructor's Dell laptop. The prompt on your computer system, including the default drive and default directory, may look somewhat different depending on how your environment is set up.
8. You have to turn hardcopy of Lab 1 Report. You will need to create two directories named Lab1 and Lab2 in the root directory of your thumb drive. All Lab1 related files and directories will be stored in the Lab 1 directory and all files related to Lab2 (next lab) will be stored in the Lab2 directory. A template for Lab 1 Report is posted in folder Assignments\Labs folder on Blackboard.

## **Introduction**

Typically, a user accesses a computer through the user interface (GUI, menu-driven, or command line interface - CLI) and the application program communicates with the OS through the application programming interface (API). At times, however, the user must communicate directly with the OS via the CLI. This lab deals with the Windows CLI.



In CLI a user communicates with the OS by issuing commands. The general form of the command is shown in the Figure below. In Windows the default/current drive and the system prompt are displayed by the OS. The user responds to an active prompt by typing a command name followed by any necessary parameters. A delimiter, usually a space, separates the command from the parameters and (if there are several) the parameters from each other.



Commands may be typed in any combination of uppercase or lowercase.

Windows commands are grouped into two categories: resident and transient. Resident commands are immediately available as they reside in RAM. Transient commands need to be loaded from a disk.

## **Tutorial**

The lab is written as a tutorial. Do not simply read it. Instead work the lab on your school or personal computer that runs Windows 10 and follow along. As you read about the command, enter it and see for yourself how the computer responds. The lab was run on my Dell laptop with Windows 10 and the tutorial contains the screen shots from this environment. If you run the lab on earlier versions of Windows (7 or 8), the system response, prompt, default/current drive, and default/current directory may be somewhat different.

## **Naming Conventions**

1. If a part of a line to enter appears in **BOLD UPPERCASE** text, it is a part of the name of a Windows command, instruction, or program and has precise meaning to the system. It must be entered exactly as is.

2. Portions of commands that you enter are in **bold lowercase** or **Mixed Case**. These are the portions of the instructions that are modifiable by users. For instance, I include suggested filenames in the commands I ask you to enter. You may later use the same command in conjunction with your own files. The italics indicate the portions of the command line you can change and still have it work.
3. The steps to perform on the computer are in ***bold italic***.
4. <Enter> means that you need to press the Enter key to execute a command.

## Help

Whenever you need help on a command description and syntax, type `HELP COMMAND_NAME` or `COMMAND_NAME /?` For example, to get help on a command `COPY`, type in lower or upper case from the command prompt:

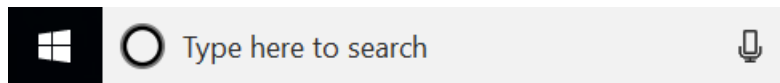
**HELP COPY**                      or                      **COPY /?**                      <Enter>

## Getting Started

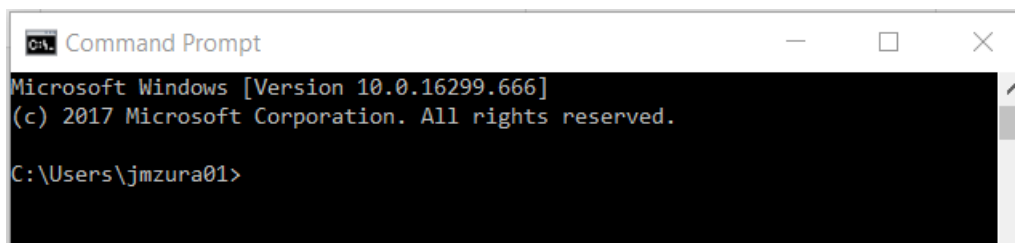
Perform these steps on the computer.

*Turn the computer on and let Windows boot.*

*Type CMD (lowercase is fine) in the open window at the bottom of the pull-down menu and <Enter>.*



You may also click on the Start button, select Windows System, and on the submenu select the Command Prompt. You should see the following screen with the initial Windows command prompt.



You can see the system prompt. On my laptop the system prompt is "C:\Users\jnzura01>", where C: is the current/default drive. On most systems "C:" indicates the hard drive, "\Users\jnzura01" is the name of the current directory, and the greater than ">" symbol is the prompt which indicates that Windows is waiting for you to enter a command.

Again, on your machine, the prompt will be different, so do not get confused. For example, you will not see the user id "jnzura01" subdirectory on your machine. However, the precise wording of the prompt is not significant to the rest of this tutorial.

You can type commands and file names in lowercase or uppercase, or mixed case because Windows is not case sensitive.

What is the meaning of the default/current drive and the default/current directory?

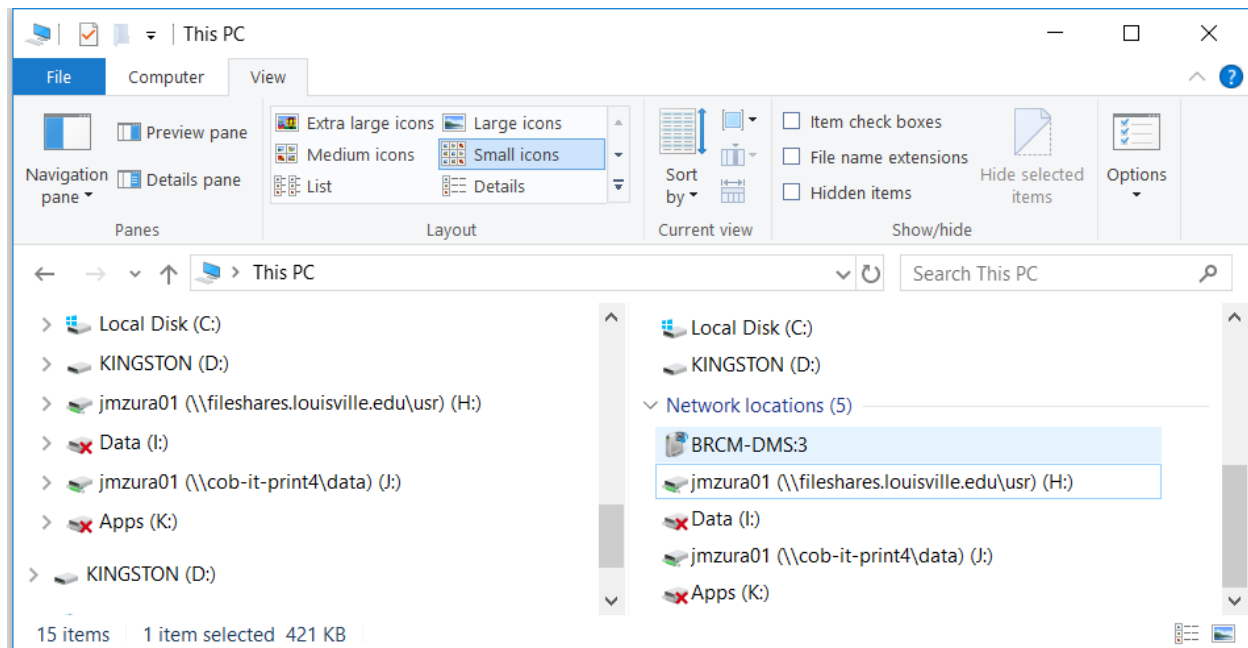
When a user types a command from the system prompt, Windows will be looking for the command on the current drive in the current directory, unless a path to the command is provided in the AUTOEXEC.BAT file or in the Windows registry system. When a command refers to some files to act upon, Windows will be looking for the files on the current drive in the current directory, unless a path to the files is provided on the command line.

Now do the following steps:

***Insert a thumb drive, preferably blank, into one of the USB ports of your computer.***

***Use the Windows Explorer to find out the letter that assigned to your thumb drive.***

What letter represents your thumb drive in the This PC window? It is likely to be either the D, E or F letter. The screen shot below is from the Windows Explorer or This PC.



The screen above shows the This PC window of my Dell laptop. One can see that my hard drive is represented by letter C and the thumb drive on my laptop is represented by letter D. The four other disk drives H, I, J, and K are the server drives on the College's network.

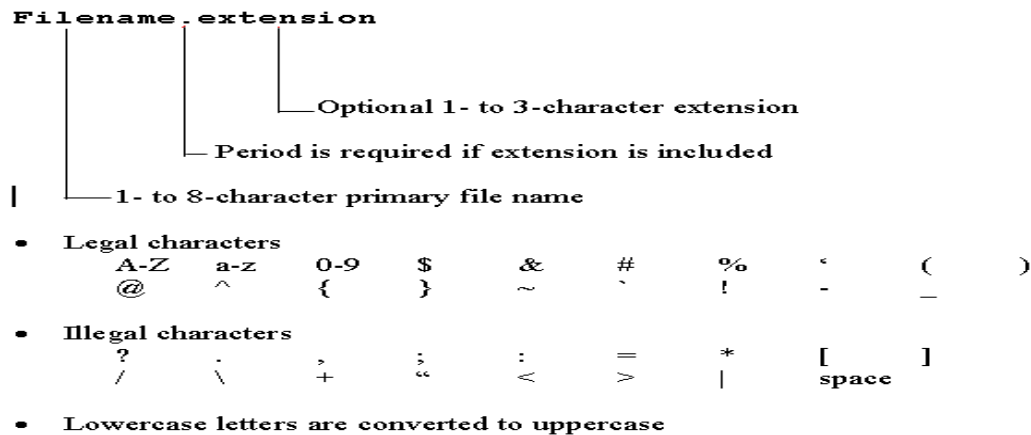
It is very likely that on your machine, letter E or F will be used for your thumb drive. If so, all references to the D drive in this tutorial will have to be replaced by letter E, F or another letter designation that represents the thumb drive on your machine!

## **Formatting**

The FORMAT command is used to format a disk, thumb drive, or another storage medium to prepare it to store files. If you were to format the thumb drive, you would now type FORMAT D: and press Enter. Do not do it, however, as most thumb drives are purchased preformatted. FORMAT is a destructive command. When you format a medium such as thumb drive, you erase whatever data might be stored on it. Do not under any circumstances format your hard drive (C drive). You could destroy your system.

## File System

The Windows file system allows a user to identify, save, and retrieve files by name. The table below is of historic significance as it presents the rules for defining the filename in old MS-DOS. A filename is composed of the name itself and an optional extension. The name consists of from 1 to 8 characters. The file name is separated from its optional, 1- to 3-character extension by a period. This is commonly known as an 8.3 file name. Some file extensions have special meaning to the OS. Several conventional extensions are: COM (command file), EXE (executable file), DVD (device driver), DAT (ASCII data file), OBJ (object module), and BAT (batch file). File names such as CON (console), COM1, COM2 (serial communication ports), PRN, LPT1, LPT2 (system printers), NUL, and CLOCK are forbidden as they are reserved for system device names.



The Windows FAT and NTFS file systems are not limited to 8.3 file names, because they have long file name support (up 257 characters or even more), but they still support the 8.3 version of long file names. Windows supports the same file extensions and adds many more. It does not allow one to use the above system device names for file names either. Also, in file names you may not use illegal characters listed above.

## Directories

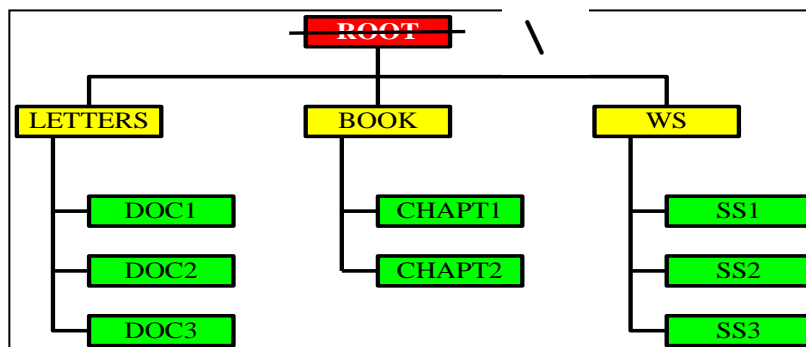
Like Unix/Linux, Windows supports a flexible upside-down tree directory structure with the root directory at the top. When the disk is formatted only a single root directory is created on disk. There is nothing peculiar about the root directory. It is just the top directory and the origin for other subdirectories which one can build under the root. Directories help to organize files. A directory is a special type of a file and the same file naming convention applies to naming directories.

Directory management is a key Windows file system function. The first time a file is written to disk, its name, starting address, creation date, and other information are recorded in the disk's directory. Later, when the file is retrieved, the OS reads the directory and searches it for file name. When a file is modified, the file system updates its directory entry. When the file is deleted, its directory entry is marked as deleted. The directory is stored on disk, but it must be transferred from disk to memory when it is searched or updated by the file system. Apart from managing disk directory, the file system also manages the file/disk allocation table which contains links/pointers to tracks and sectors in which the rest of the file is stored on disk. When MS-DOS was the primary OS, the disk capacity it managed was rather small; it amounted to hundreds of MBs. As a result, a file allocation table (FAT) was implemented by a set of single and direct pointers/links to a file. Recent versions of Windows manage huge disk space (hundreds of GB or TB). As a result, file systems use much more

complicated FAT which access files by a chain of several indirect pointers, three or even four levels of pointers. We will revisit FAT when we cover the file management in class.

## Subdirectories

When a disk is formatted, a single root directory is created by the format routine. Using a single directory is fine for a few files, but as the number of files increases, distinguishing them becomes increasingly difficult. For example, imagine a work disk that holds several different types of files. To simplify keeping track of the files, Windows allows the user to create special files called subdirectories. For example, the figure below shows the root directory and three subdirectories. LETTERS hold letters and other correspondence. A book's chapters are stored under subdirectory BOOK. Finally spreadsheets are grouped in subdirectory WS. Think of subdirectories as a file folder that allows you to group related files and thus organize the disk. The word ROOT representing the root directory is used here for illustration purposes only. In Windows commands the root directory is represented by the backslash \ (it is the first backslash in a path), and you will never actually see the word ROOT in actual commands.



## Path Names

When subdirectories are used, you need more than a simple file name to find a file. For example, it is possible to have files named PAY stored in two different subdirectories. A reference to PAY would thus be ambiguous – which PAY do you mean?

To fully identify a file you need a complete path name. For example,

\LETTERS\PAY

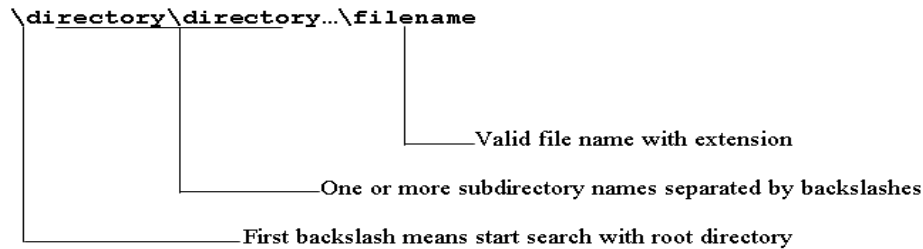
and

\WS\PAY

are two different files. Assume that the first one is stored in subdirectory LETTERS, whereas the second one is stored in subdirectory WS.

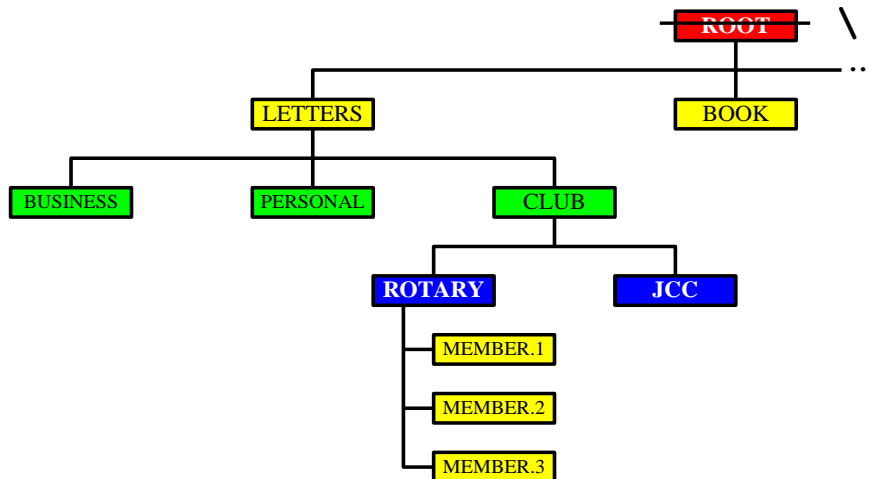
Study the path names listed above. The first backslash \ references the root directory. The second backslash separates the subdirectory name from the file name. The first path name shown above tells Windows to start with the root directory, find a subdirectory named LETTERS, and search the subdirectory for a file named PAY. The second path name tells Windows to start with the root directory, find a subdirectory named WS, and search the subdirectory for a file named PAY.

The rules for defining a path name in MS-DOS are in the table below. Windows has the same rules, but a path name can be much longer and may contain up to 260 characters.



- To define a subdirectory name, use the rules for defining a file name with no extension.
- Maximum path name length is 63 characters.
- If path name does not start with a backslash, search begins with current working directory.

It is possible to divide a subdirectory into lower-level subdirectories. For example, the figure below shows LETTERS broken into three subdirectories. One, CLUB, is further subdivided into ROTARY and JCC. To retrieve a document named MEMBER.3 from the ROTARY subdirectory, the absolute/full path name would be: \LETTERS\CLUB\ROTARY\MEMBER.3. However, being in already in the CLUB directory, you can specify the relative path ROTARY\MEMBER.3 to access file MEMBER.3. Windows will then internally build the absolute/full path to access the file. Again the word ROOT in the diagram below is only used for illustration. So note that in the path \LETTERS\CLUB\ROTARY\MEMBER.3 the first backslash "\" represents the root directory.



Note how the path name leads from directory to directory until you reach the desired file. One needs to distinguish between a file name and a path name. A file name is the name assigned to a file, whereas a path name defines the series of directories that lead to the desired file.

### Viewing a Directory

Before you begin creating and manipulating your own directories, it might be wise to look through existing ones.

This is the screen which you would see after executing the **CMD** command on p. 4.

```

C:\> Command Prompt
Microsoft Windows [Version 10.0.16299.666]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\jmzura01>

```



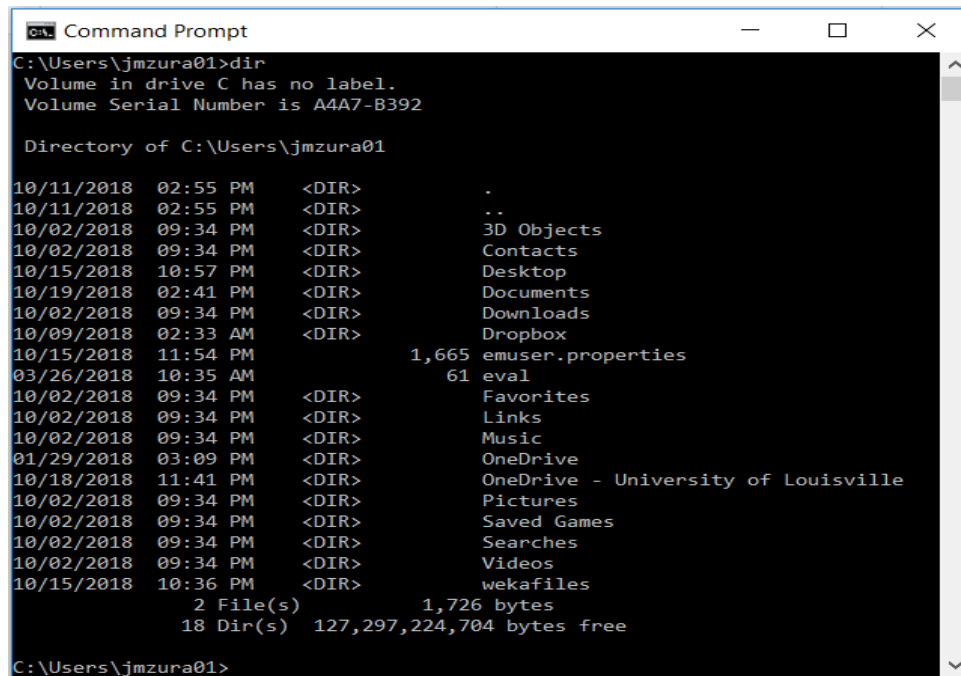
Again, on my computer the current directory on the C drive is "\\Users\\jmzura01". On your machine the prompt will be different, so do not get confused.

To display the contents of the current subdirectory on your C drive, type:

**DIR**                      <Enter>

The DIR command lists filenames and names of subdirectories on the screen. Note that the attributes such as the date and time when a file (directory is also a file) was created/modified is displayed as well as the file names, file sizes in bytes, and the overall statistics.

The screen you would see may be similar to this one.



```
C:\Users\jmzura01>dir
Volume in drive C has no label.
Volume Serial Number is A4A7-B392

Directory of C:\Users\jmzura01

10/11/2018  02:55 PM  <DIR>          .
10/11/2018  02:55 PM  <DIR>          ..
10/02/2018  09:34 PM  <DIR>          3D Objects
10/02/2018  09:34 PM  <DIR>          Contacts
10/15/2018  10:57 PM  <DIR>          Desktop
10/19/2018  02:41 PM  <DIR>          Documents
10/02/2018  09:34 PM  <DIR>          Downloads
10/09/2018  02:33 AM  <DIR>          Dropbox
10/15/2018  11:54 PM             1,665 emuser.properties
03/26/2018  10:35 AM             61    eval
10/02/2018  09:34 PM  <DIR>          Favorites
10/02/2018  09:34 PM  <DIR>          Links
10/02/2018  09:34 PM  <DIR>          Music
01/29/2018  03:09 PM  <DIR>          OneDrive
10/18/2018  11:41 PM  <DIR>          OneDrive - University of Louisville
10/02/2018  09:34 PM  <DIR>          Pictures
10/02/2018  09:34 PM  <DIR>          Saved Games
10/02/2018  09:34 PM  <DIR>          Searches
10/02/2018  09:34 PM  <DIR>          Videos
10/15/2018  10:36 PM  <DIR>          wekafiles
                2 File(s)              1,726 bytes
               18 Dir(s)  127,297,224 bytes free

C:\Users\jmzura01>
```

All versions of Windows contain a directory named *Windows* that holds numerous system files.

To list the contents of the *Windows* directory, type:

**DIR \WINDOWS**                      <Enter>

DIR is the command name and \WINDOWS (note the initial backslash) is the path name to the *Windows* directory which stems from the root "\\" directory.

The screen you might see will be similar to the following. A list of file names and other information will scroll rapidly off your screen. The figure shows the last screen followed by a new prompt.

```

C:\Users\jzmzura01>DIR
09/29/2017 09:46 AM <DIR> SystemResources
10/19/2018 11:42 PM <DIR> SysWOW64
09/29/2017 09:46 AM <DIR> TAPI
02/13/2018 03:53 PM <DIR> Tasks
10/19/2018 11:58 PM <DIR> Temp
10/02/2018 09:30 PM <DIR> TextInput
09/29/2017 09:46 AM <DIR> tracing
09/29/2017 09:46 AM <DIR> twain_32
09/29/2017 09:42 AM 65,536 twain_32.dll
09/29/2017 09:46 AM <DIR> Vss
09/29/2017 09:46 AM <DIR> Web
10/19/2018 11:08 PM 167 win.ini
10/20/2018 12:04 AM 276 WindowsUpdate.log
09/29/2017 09:42 AM 11,776 winhlp32.exe
10/19/2018 11:42 PM <DIR> WinSxS
09/29/2017 09:41 AM 316,640 WMSysPr9.prx
09/29/2017 09:41 AM 11,264 write.exe
31 File(s) 11,063,804 bytes
78 Dir(s) 125,557,039,104 bytes free
C:\Users\jzmzura01>

```

As entered, the directory command was not very useful because the list of files in directory *Windows* is much too big to fit on a single screen. Type the command

**DIR \WINDOWS /P**      <Enter>

A space separates the command from the parameter, and a slash is a regular slash, not a backslash. A "P" switch stands for pause and prevents the screen from scrolling. A single page of information appears on your screen. See the screen shot below. One can see that the commands can be typed in lower or upper case or mixed case.

```

C:\Users\jzmzura01>DIR \WINDOWS /P
Volume in drive C has no label.
Volume Serial Number is A4A7-B392

Directory of C:\WINDOWS

10/04/2018 01:09 PM <DIR> .
10/04/2018 01:09 PM <DIR> ..
09/29/2017 09:46 AM <DIR> addins
01/28/2018 05:39 PM <DIR> AdminArsenal
02/14/2018 11:03 AM <DIR> appcompat
10/02/2018 09:29 PM <DIR> apppatch
10/17/2018 09:17 AM <DIR> AppReadiness
02/15/2018 07:17 PM <DIR> bcasdvr
09/29/2017 09:41 AM 65,536 bfsvc.exe
09/29/2017 09:46 AM <DIR> Boot
09/29/2017 09:46 AM <DIR> Branding
10/19/2018 11:26 PM <DIR> CbsTemp
05/30/2018 03:23 PM <DIR> CCM
10/19/2018 11:16 PM <DIR> ccmcache
02/27/2018 12:11 PM <DIR> ccmsetup
Press any key to continue . . .

```

To reiterate, reading from left to right, each Windows 10 line display's a file's creation date and time, either <DIR> (for directory) name or the file size in bytes, and the file name with an extension, if exists. The last line reads *Press any key to continue*.

***Press the space bar or any key.***

and a new screen of directories and file names will appear. Continue pressing the space bar to step through the directory one screen at a time until you see the prompt.

Type:

**DIR \WINDOWS | MORE**

<Enter>

The command above uses a pipe "|". The command has a similar affect to the previous command using an option "/P". We implemented here a piping operation. Simply, the output from the DIR command is used as the input to the MORE command. The latter command prevents the output from scrolling off the screen and it displays one screen at a time.

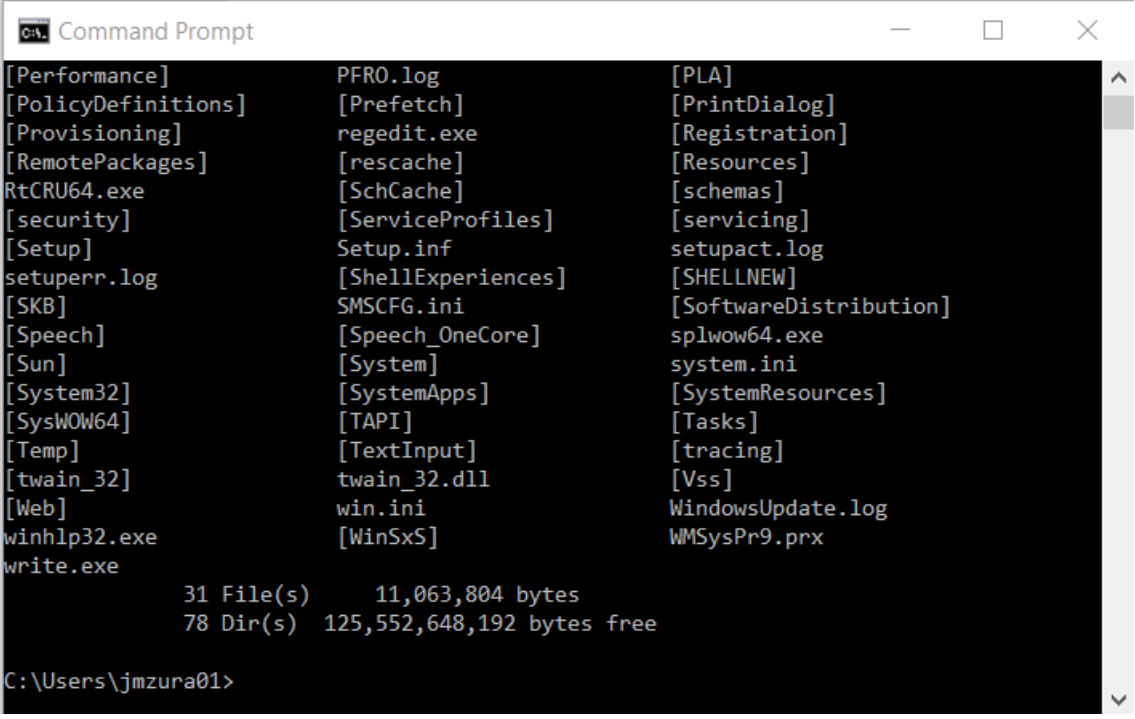
*Continue pressing the space bar* to step through the directory one screen at a time until you see the prompt. We will discuss piping more thoroughly later in the lab.

A wide mode display, switch "W", is another useful option. From the prompt, type:

**DIR \WINDOWS /W**

<Enter>

The resulting screen below lists only the directories names in [ ] and file names across the screen. The Windows directory contains so many files that not even a wide mode list will fit on a single screen. Note that other file attributes such as the file size and date are suppressed.



```
C:\> Command Prompt

[Performance]      PFR0.log          [PLA]
[PolicyDefinitions] [Prefetch]        [PrintDialog]
[Provisioning]      regedit.exe       [Registration]
[RemotePackages]    [rescache]        [Resources]
RtCRU64.exe         [SchCache]        [schemas]
[security]          [ServiceProfiles] [servicing]
[Setup]            Setup.inf         setupact.log
setuperr.log        [ShellExperiences] [SHELLNEW]
[SKB]              SMSCFG.ini        [SoftwareDistribution]
[Speech]           [Speech_OneCore]  splwow64.exe
[Sun]              [System]          system.ini
[System32]         [SystemApps]      [SystemResources]
[SysWOW64]         [TAPI]            [Tasks]
[Temp]            [TextInput]       [tracing]
[twain_32]         twain_32.dll      [Vss]
[Web]             win.ini          WindowsUpdate.log
winhlp32.exe       [WinSxS]          WMSysPr9.prx
write.exe

                31 File(s)      11,063,804 bytes
                78 Dir(s) 125,552,648,192 bytes free

C:\Users\jnzura01>
```

Now let's try the TREE command which displays directory paths and subdirectories, in other words it displays the hierarchy of directories. The TREE command may not work, if you access Windows command line on VMware in the CIS/COB lab. In addition to displaying directories, the TREE/F command allows one to display files within directories. An "F" is a switch.

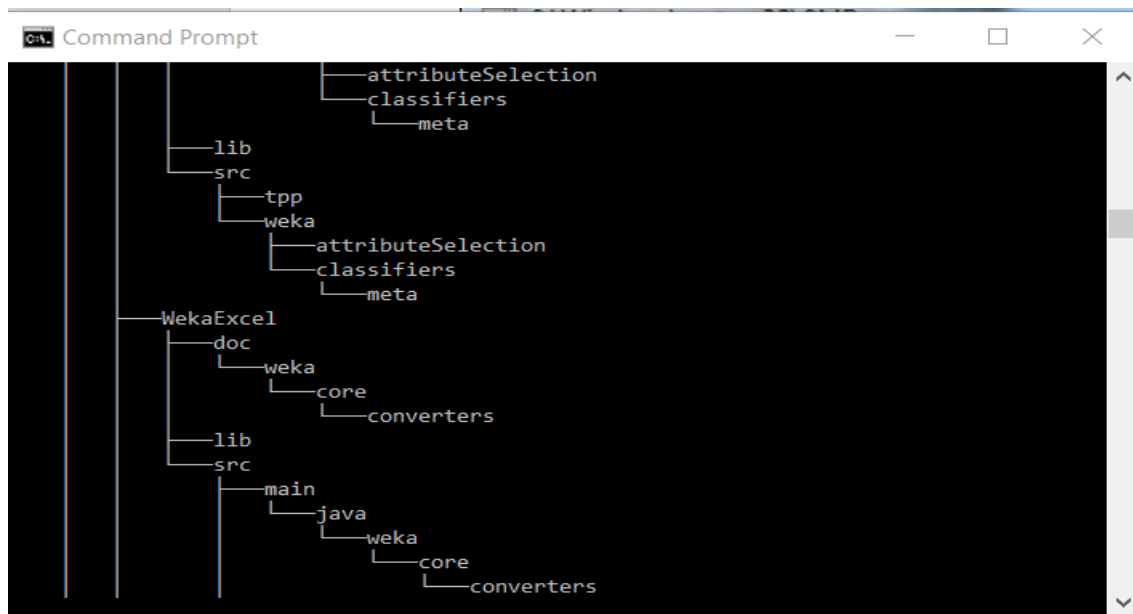
Pick out a directory on your C drive in which you have many subdirectories and files. The "Users\jnzura01\" directory would be good on my machine. On yours, it may be a different directory.

Type:

**TREE**

<Enter>

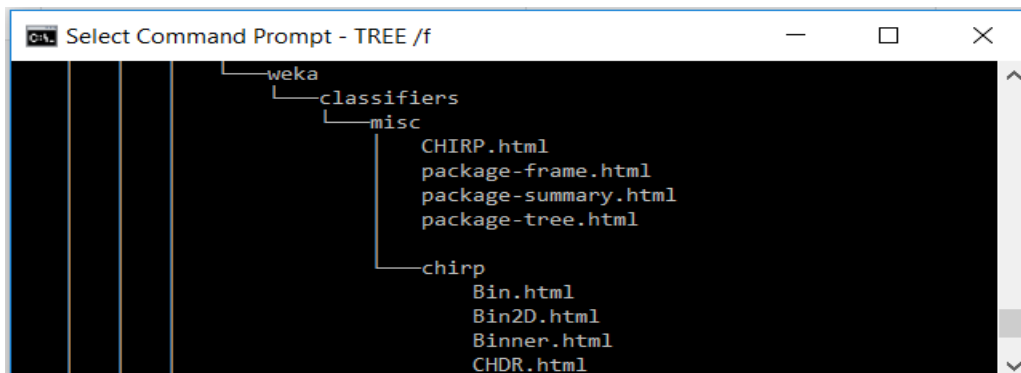
The command displayed the directory structure/hierarchy in the "Users\jnzura01" directory and subdirectories under it. As I have many subdirectories under the "Users\jnzura01" directory, the output scrolled off the screen rapidly when I ran this command. The figure below shows one of the many screen displayed. You may run this command by typing TREE | MORE. If you want to interrupt the command, press Ctrl-C (hold down the Ctrl key and press C).



Type:

**TREE /F**                      <Enter>

You should see on the computer screen the structure/hierarchy of the subdirectories and all files (the F switch/ option) stored in the "Users\jnzura01" directory and subdirectories under it. The F switch displays files within each subdirectory. The example screen follows.



For the time being we are done with the directories and the TREE command. Make sure that the thumb drive is in one of the USB ports. From the "C:\Users\jnzura01>" prompt, type the letter D followed directly by a colon ":"

**D:**                                      <Enter>

Again, on your machine the thumb drive may be represented letter E or F. If so, replace all references to D's with E's or F's in all commands that you encounter in this tutorial. From now on, the prompt on your laptop and mine should match, perhaps except the letter representing the thumb drive.

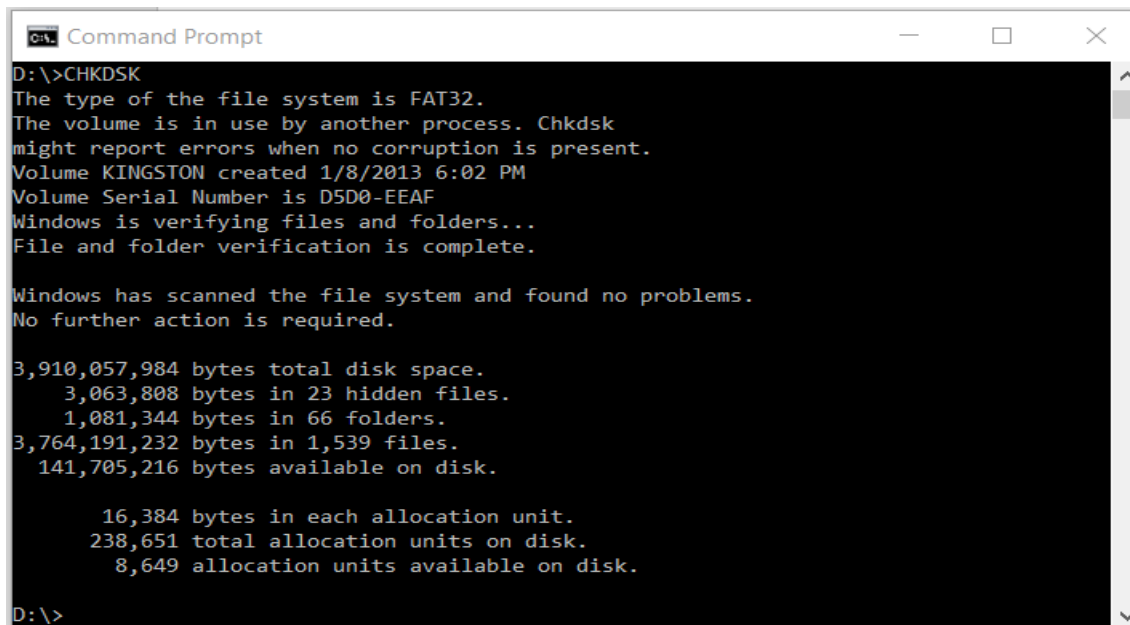
The prompt should read.



```
C:\Users\jnzura01>D:
D:\>
```

The CHKDSK command checks a disk directory and reports its contents, checks the size of RAM too. Type:

**CHKDSK**                      <Enter>



```
D:\>CHKDSK
The type of the file system is FAT32.
The volume is in use by another process. Chkdsk
might report errors when no corruption is present.
Volume KINGSTON created 1/8/2013 6:02 PM
Volume Serial Number is D5D0-EEAF
Windows is verifying files and folders...
File and folder verification is complete.

Windows has scanned the file system and found no problems.
No further action is required.

3,910,057,984 bytes total disk space.
 3,063,808 bytes in 23 hidden files.
 1,081,344 bytes in 66 folders.
3,764,191,232 bytes in 1,539 files.
 141,705,216 bytes available on disk.

    16,384 bytes in each allocation unit.
 238,651 total allocation units on disk.
   8,649 allocation units available on disk.

D:\>
```

Note that the CHKDSK command displayed the status of thumb drive D, which is the current drive. The command acts on this drive.

### **Creating, Changing, and Removing Directories**

One can use the following commands to create directory (MKDIR or MD), change directory (CHDIR or CD), and remove directory (RMDIR or RD).

For example:

```
D:\>MD LETTERS    – creates directory LETTERS within the root directory
D:\>MD WS         – creates directory WS within the root directory
D:\>RD WS         – removes directory WS
D:\>CD\LETTERS    – changes the default current directory to LETTERS
                  D:\LETTERS> will be displayed as the prompt.
```

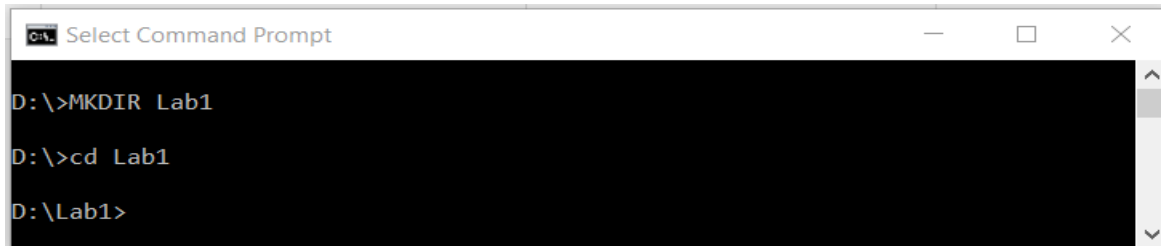
The "D:\>" prompt indicates that the D drive is the current drive and the current directory is the root directory "\". So the above commands concern the D drive.

On your thumb drive you are about to create a new directory Lab1, and three new subdirectories and several files within one of the subdirectories. In case you have some other directories and data files on your thumb drive, it's OK. We simply want to keep all Lab1 related work in the Lab1 directory. Similarly, all Lab 2 related work will be kept in the Lab2 directory. See Lab 2.

Use the MKDIR (or MD - shortcut) command to create directory Lab1. Type:

Type: **MKDIR Lab1** <Enter>  
**CD Lab1** or **CD \Lab1** <Enter>

Your screen should be like the one below.



```

Select Command Prompt

D:\>MKDIR Lab1

D:\>cd Lab1

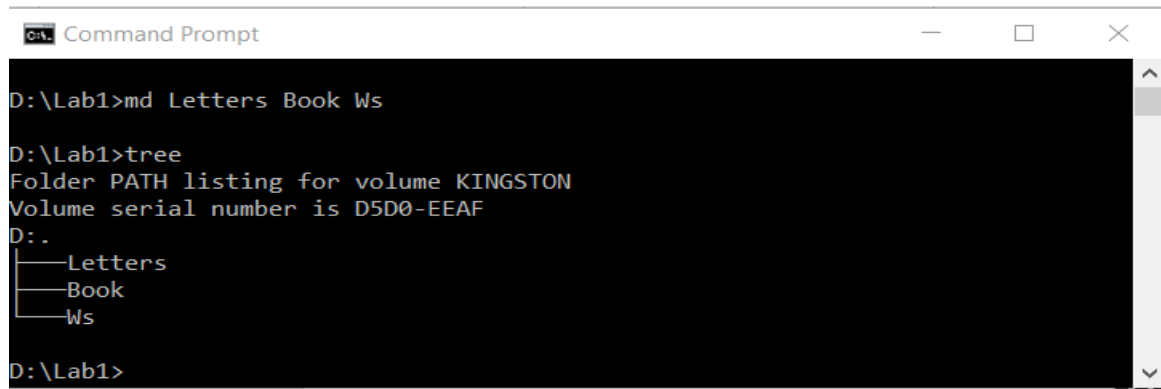
D:\Lab1>
```

Your current drive and current directory are D and Lab1, respectively.

Now, under the Lab1 directory create three directories: Letters, Book, and Ws. You could do it with three separate commands: MD Letters <Enter>, MD Book <Enter>, and MD Ws <Enter>. However, we will create the three directories with one command. Type:

**MD Letters Book Ws** <Enter>  
**TREE** <Enter>

Note that there should be at least one blank space between the command itself and the name of the first directory Letters and between the two remaining directories.



```

Command Prompt

D:\Lab1>md Letters Book Ws

D:\Lab1>tree
Folder PATH listing for volume KINGSTON
Volume serial number is D5D0-EEAF
D:
|_ Letters
|_ Book
|_ Ws

D:\Lab1>
```

To traverse to *Letters* subdirectory, type:

**CD \Lab1\Letters** (absolute path) or just **CD Letters** (relative path) <Enter>

Now, you should see prompt D:\Lab1\Letters>.

### Creating Files with NOTEPAD and COPY Command

One can use the Windows Notepad or COPY command to create files. As a COPY command is somewhat awkward, we will use Notepad most of the time.

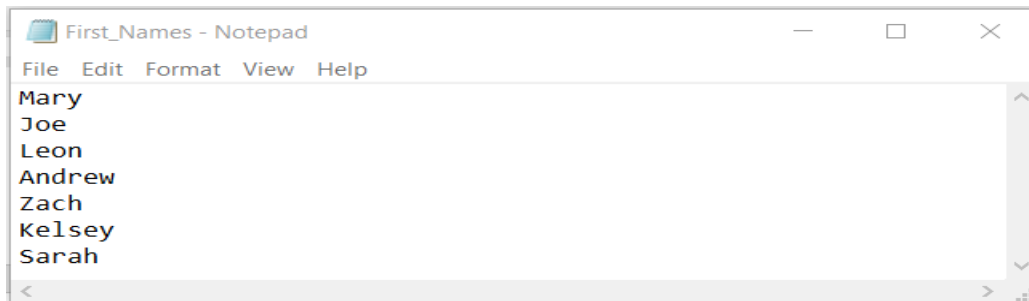
From the D:\Lab1\Letters> prompt, type:

**NOTEPAD**

<Enter>

This command opens the Windows Notepad.

Type the following 7 first names which you see in the Notepad Window below and save them in a text file (extension .txt) named *First\_Names* in the *Letters* subdirectory.



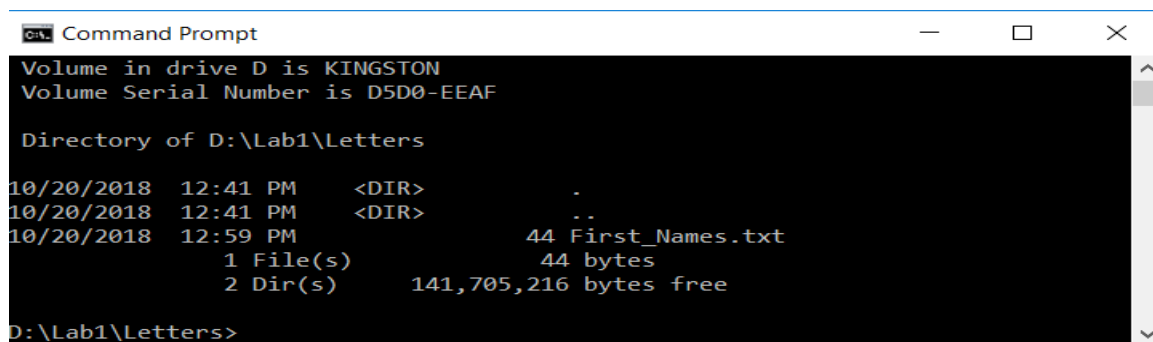
Note that file name *First\_Names* contains 11 characters. As mentioned in Windows 10 file names can contain up to 257 characters or more. The exact number of characters depends on the file system such as NTFS.

Type:

**DIR**

<Enter>

to see if file *First\_Names.txt* is indeed in the subdirectory "*\Lab1\Letters*".



Now, you need to perform the following steps on your own.

***Using Notepad create two more files named Sally, and Tom in the subdirectory Letters, a file named Asgn1 in the subdirectory Book, and a file named Chap1 in the subdirectory Ws.***

***Type in each of the four files any two short sentences of your own.***

A path to traverse to the Book directory is *\Lab1\Book*, and a path to move to subdirectory Ws is *\Lab1\Ws*.

After you perform the above steps, type:

**CD ..**

<Enter>

The command above should bring you to subdirectory *Lab1*, which is the parent directory for subdirectories *Letters*, *Book*, and *Ws*. Simply, "CD .." always moves you up one level in the directory structure. In other words, it moves you to the parent directory.

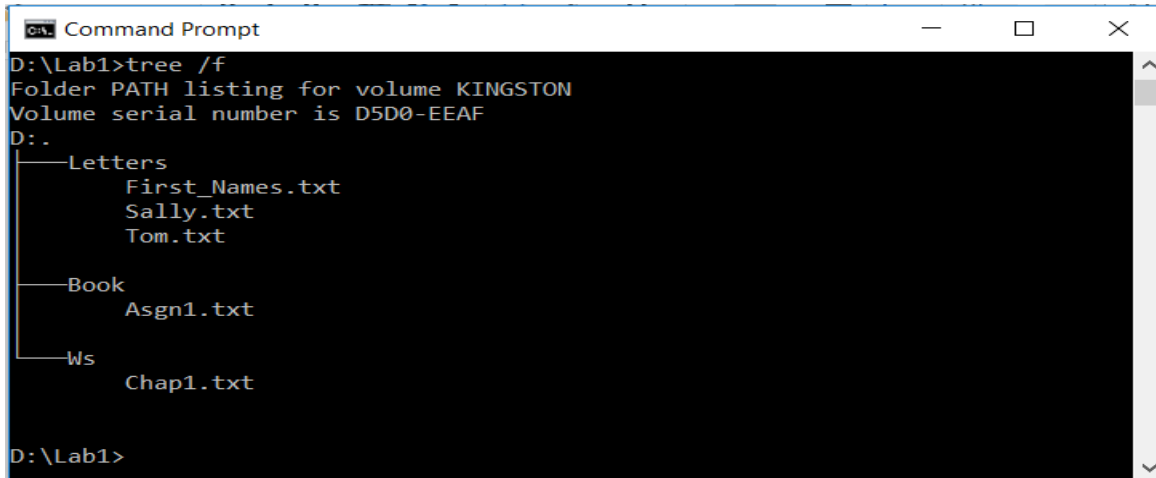
Type:

**TREE /F**

<Enter>

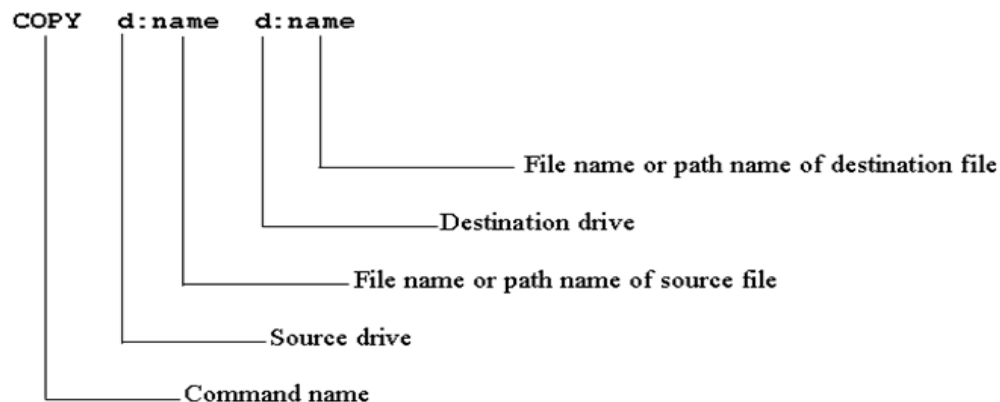
To reiterate, the F switch displays files within each subdirectory. It is not related to letter F that may represent your thumb drive on your computer.

This is the screen which you should see.



```
CA: Command Prompt
D:\Lab1>tree /f
Folder PATH listing for volume KINGSTON
Volume serial number is D5D0-EEAF
D:..
├── Letters
│   ├── First_Names.txt
│   ├── Sally.txt
│   └── Tom.txt
├── Book
│   └── Asgn1.txt
└── Ws
    └── Chap1.txt

D:\Lab1>
```



- If no destination file name is given, the source file name is used and the drives must be different.
- The source file designation must differ in some way (file name, drive, and/or directory) from the destination file specification.

Let's see now how a COPY command can be used to create files. When Windows carries out a COPY command, it reads the file specified in the first parameter (the source file) and copies it to the file specified in the second parameter (the destination file). For syntax of the COPY command see the table below. A simple way to create a short file is to copy it from console (your system's keyboard and display). We will create one more file named *Jim* in the subdirectory *Letters*.

Type the command:

**COPY CON \Lab1\Letters\Jim**

<Enter>

The first (source) file name, CON, stands, for console. The second (destination) file name specifies a full/absolute path name. The destination file will be stored on drive D (thumb drive) subdirectory *Letters* and assigned file name *Jim*. The following command with a relative path should also work: COPY CON *Letters\Jim* because you should be in the Lab1 subdirectory.



The cursor will appear directly under the command line (you will see no prompt). Type:

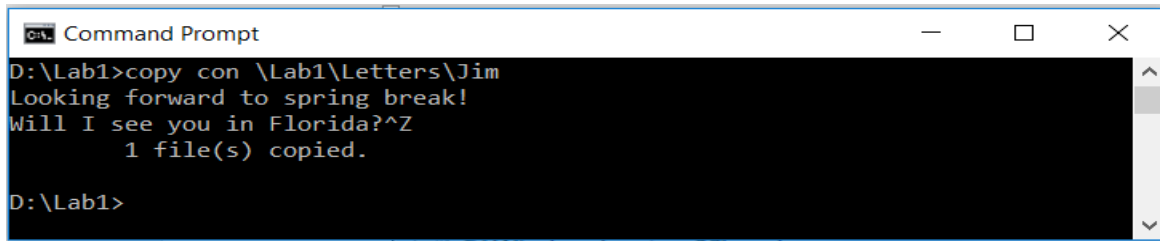
*Looking forward to spring break!* <Enter>

*Will I see you in Florida?* <Enter>

Now hold down Control (*Ctrl*) key *and* press *Z* <Enter>

The system will generate two characters "^Z", under the first two lines you have typed. You should also see the message confirming that the file was copied and the prompt.

This is what you should see on the screen.



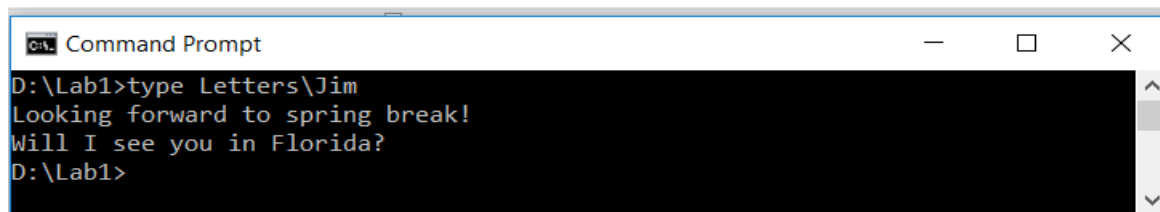
```
Command Prompt
D:\Lab1>copy con \Lab1\Letters\Jim
Looking forward to spring break!
Will I see you in Florida?^Z
        1 file(s) copied.

D:\Lab1>
```

Using the COPY CON command for data entry is very inconvenient as it does not allow you to modify the previous lines you have typed. It is an awkward remnant from MS-DOS.

To display the contents of file *Jim*, type:

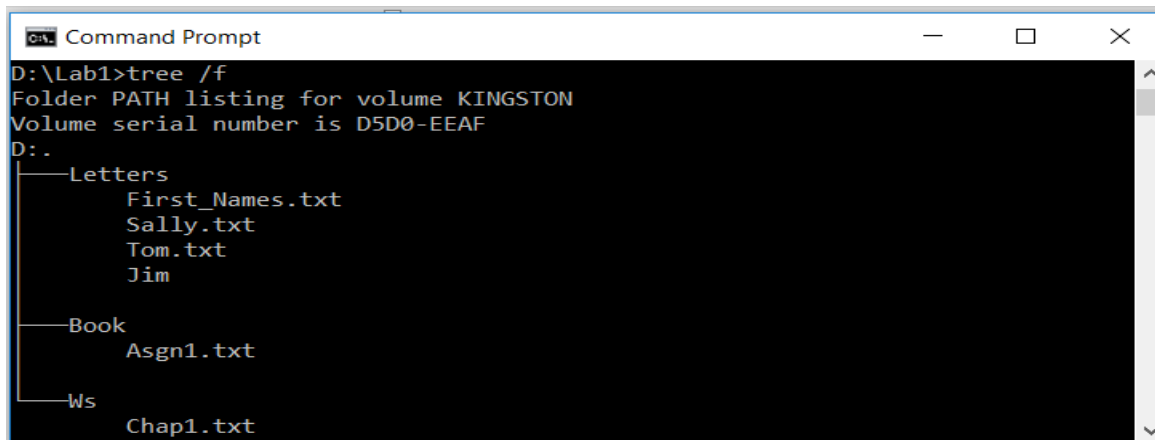
**TYPE \Lab1\Letters\Jim** or **TYPE Letters\Jim** <Enter>



```
Command Prompt
D:\Lab1>type Letters\Jim
Looking forward to spring break!
Will I see you in Florida?
D:\Lab1>
```

Type:

**TREE /F** <Enter>



```
Command Prompt
D:\Lab1>tree /f
Folder PATH listing for volume KINGSTON
Volume serial number is D5D0-EEAF
D:.
├── Letters
│   ├── First_Names.txt
│   ├── Sally.txt
│   ├── Tom.txt
│   └── Jim
├── Book
│   └── Asgn1.txt
└── Ws
    └── Chap1.txt
```

## Redirecting Input and Output

Windows commands assume the standard input from the keyboard and route the output to the monitor screen. Input, however, can come from a file rather than from the keyboard; output can

go to a file or to a printer instead of to a screen. Let's first discuss these concepts without using a computer.

#### Redirecting Output - ">" or ">>" signs

The commands *DIR > PRN* or *DIR > LPT1* lists the contents of the directory and sends the output to the system printer attached to the *LPT1* port. In a network environment these two commands may not work.

The command *DIR > Dirlis.dat* lists the contents of the directory and sends the output to a file called *Dirlis.dat*. If the file does not exist, Windows creates it. If the file exists, Windows overwrites it.

The command *DIR >> Dirlist.dat* lists the contents of the directory and appends (">>") the output to a file called *Dirlist.dat*. If the file does not exist, it will be created. If the file exists, the directory list will be appended to file the existing contents of file *Dirlist.dat*.

#### Redirecting Input - "<" sign

The command *SORT < Lst.dat* sorts in the ascending order the data coming from file *Lst.dat* and displays the result on the screen.

#### Redirecting Input and Output

The command *SORT /R < Lst.dat > Lst1.dat* sorts in the descending order the data (lines) coming from file *Lst.dat* and routes the sorted file to *Lst1.dat*. File *Lst1.dat* will not be displayed on the screen.

Type:

**CD \Lab1\Letters**      or      **CD Letters**      <Enter>

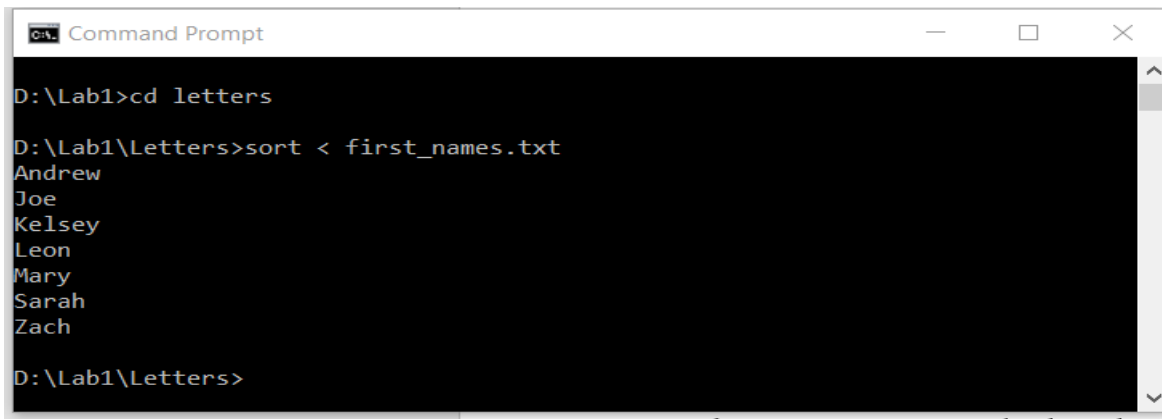
You should be in subdirectory *Letters* now. The first command above uses a full path starting with the root directory \, whereas the second command uses relative path *Letters* as the current subdirectory is already *\Lab1*. When a relative path is used in the command, Windows internally concatenates *\Lab1* with *Letters* to create full path *\Lab1\Letters* to locate the *Letters* subdirectory.

Type:

**SORT < First\_Names.txt**      <Enter>

The SORT command takes the input from a file *First\_Names.txt*, sorts first names in the ascending order in a temporary file, and displays the sorted names on the computer screen. The temporary file used for sorting is erased.

Your computer screen should look as follows.



```
Command Prompt
D:\Lab1>cd letters
D:\Lab1\Letters>sort < first_names.txt
Andrew
Joe
Kelsey
Leon
Mary
Sarah
Zach
D:\Lab1\Letters>
```

Type:

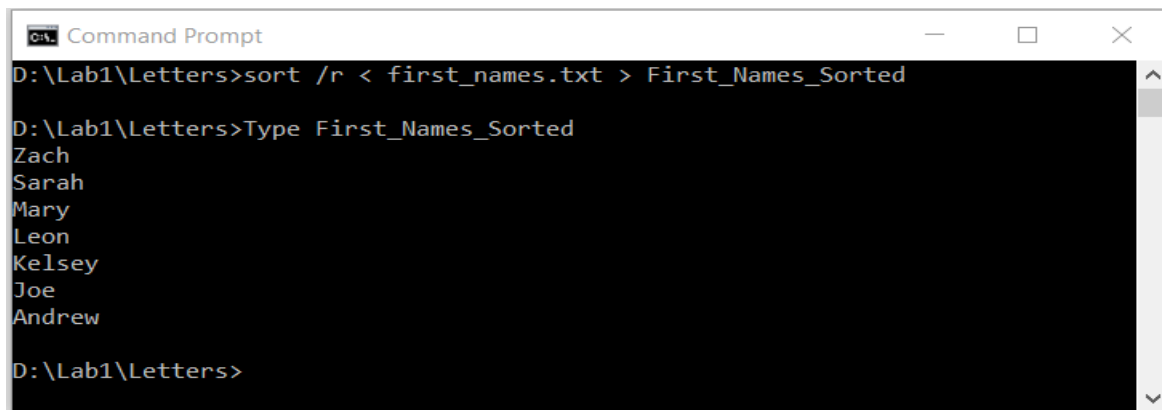
**`SORT /R < First_Names.txt > First_Names_Sorted`** <Enter>

The command `sort` takes the input from a file named *First\_Names.txt*, performs sorting in the descending order (option `/R`) in a temporary file and routes the sorted output to a file named *First\_Names\_Sorted*. If the latter file already exists it will be overwritten. If file *First\_Names\_Sorted* does not exist, it will be created. The contents of file *First\_Names.txt* does not change.

Note that the contents of file *First\_Names\_Sorted*, which contains sorted first names in the descending order, is not displayed on the screen. Instead, the output from the `SORT` command has been routed to a file named *First\_Names\_Sorted*.

To display the contents of the file, type:

**`TYPE First_Names_Sorted`** <Enter>

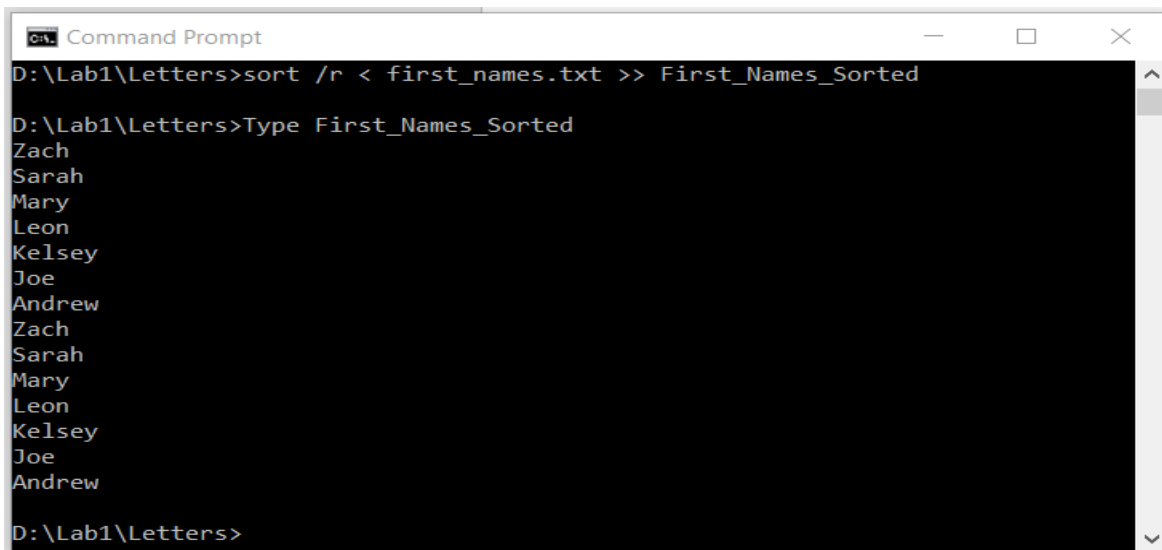


```
Command Prompt
D:\Lab1\Letters>sort /r < first_names.txt > First_Names_Sorted
D:\Lab1\Letters>Type First_Names_Sorted
Zach
Sarah
Mary
Leon
Kelsey
Joe
Andrew
D:\Lab1\Letters>
```

Type:

**`SORT /R < First_Names.txt >> First_Names_Sorted`** <Enter>

**`TYPE First_Names_Sorted`** <Enter>



```
Command Prompt
D:\Lab1\Letters>sort /r < first_names.txt >> First_Names_Sorted
D:\Lab1\Letters>Type First_Names_Sorted
Zach
Sarah
Mary
Leon
Kelsey
Joe
Andrew
Zach
Sarah
Mary
Leon
Kelsey
Joe
Andrew
D:\Lab1\Letters>
```

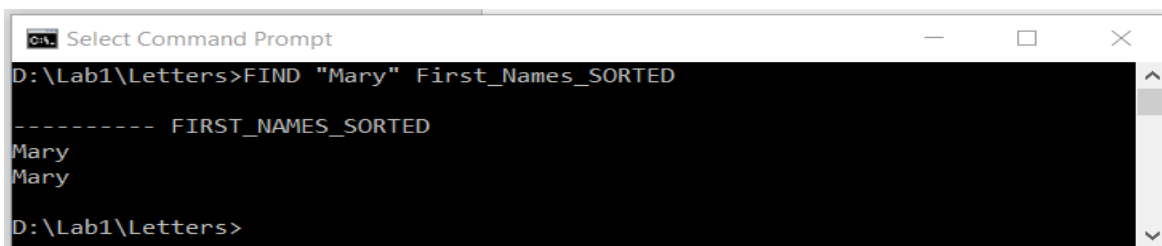
Do you see the effect of the operator ">>" append?

### **Filtering** - the FIND command

Type:

**FIND "Mary" First\_Names\_Sorted** <Enter>

Only lines with “Mary” will be displayed. Search is case sensitive. Note that it is OK to type the name of the file in mixed case like on the screen below.



```
Select Command Prompt
D:\Lab1\Letters>FIND "Mary" First_Names_SORTED
----- FIRST_NAMES_SORTED
Mary
Mary
D:\Lab1\Letters>
```

### **Piping** - "|" vertical bar

Piping can be used to make output from one source into input for another. You can pipe the standard output to another command or to a filter.

To see full effect of piping one should have many files and directories in your current drive. I do not have many files on my thumb drive, so I have to switch to the C drive. However, on your machine you may to use this command with the thumb drive, if you feel that it has meaningful amount of files. Because several next commands will be acting on the C drive, your screen and mine may be different.

Type the command:

**C:** <Enter>

You should see this now.



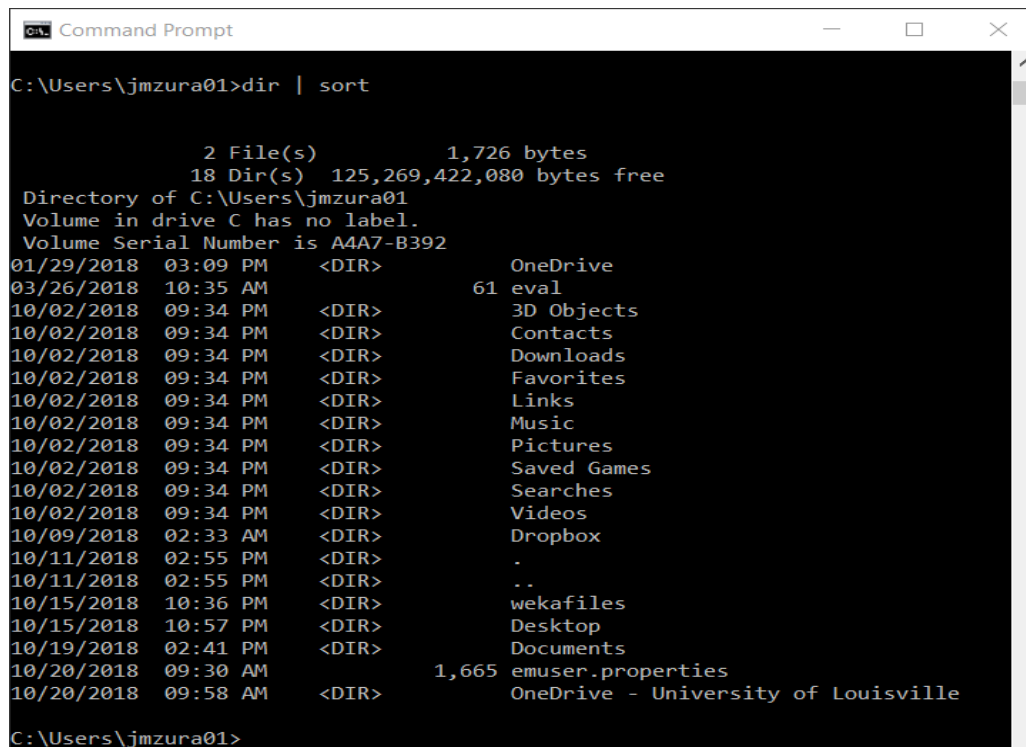
```
Command Prompt
D:\Lab1\Letters>C:
C:
```

Type:

**DIR | SORT**

<Enter>

The DIR command lists the contents of the directory in a temporary file. It then uses the temporary file as input to the SORT program, and displays the directory sorted in the ascending order by date on the screen.



```
C:\Users\jzmzura01>dir | sort

                2 File(s)                1,726 bytes
                18 Dir(s) 125,269,422,080 bytes free
Directory of C:\Users\jzmzura01
Volume in drive C has no label.
Volume Serial Number is A4A7-B392
01/29/2018  03:09 PM    <DIR>          OneDrive
03/26/2018  10:35 AM                61 eval
10/02/2018  09:34 PM    <DIR>          3D Objects
10/02/2018  09:34 PM    <DIR>          Contacts
10/02/2018  09:34 PM    <DIR>          Downloads
10/02/2018  09:34 PM    <DIR>          Favorites
10/02/2018  09:34 PM    <DIR>          Links
10/02/2018  09:34 PM    <DIR>          Music
10/02/2018  09:34 PM    <DIR>          Pictures
10/02/2018  09:34 PM    <DIR>          Saved Games
10/02/2018  09:34 PM    <DIR>          Searches
10/02/2018  09:34 PM    <DIR>          Videos
10/09/2018  02:33 AM    <DIR>          Dropbox
10/11/2018  02:55 PM    <DIR>          .
10/11/2018  02:55 PM    <DIR>          ..
10/15/2018  10:36 PM    <DIR>          wekafiles
10/15/2018  10:57 PM    <DIR>          Desktop
10/19/2018  02:41 PM    <DIR>          Documents
10/20/2018  09:30 AM                1,665 emuser.properties
10/20/2018  09:58 AM    <DIR>          OneDrive - University of Louisville

C:\Users\jzmzura01>
```

Type:

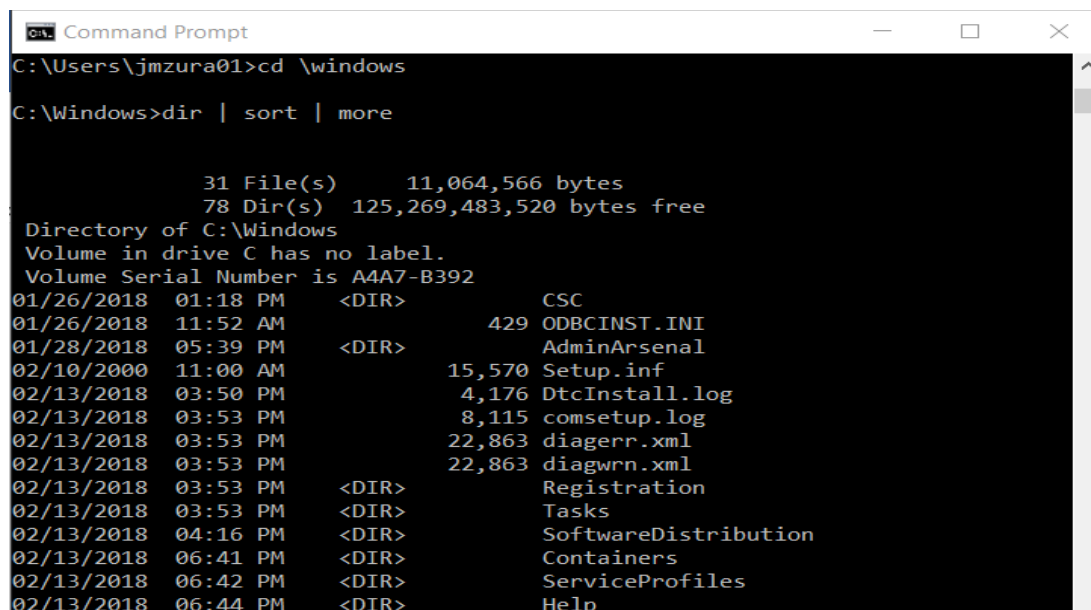
**CD \WINDOWS**

<Enter>

**DIR | SORT | MORE**

<Enter>

Convert the output of the DIR command to input for the SORT command. The output from the SORT command is then converted to input for the MORE command, and the MORE command displays one screen at a time.



```
C:\Users\jzmzura01>cd \windows
C:\Windows>dir | sort | more

                31 File(s)                11,064,566 bytes
                78 Dir(s) 125,269,483,520 bytes free
Directory of C:\Windows
Volume in drive C has no label.
Volume Serial Number is A4A7-B392
01/26/2018  01:18 PM    <DIR>          CSC
01/26/2018  11:52 AM                429 ODBCINST.INI
01/28/2018  05:39 PM    <DIR>          AdminArsenal
02/10/2000  11:00 AM                15,570 Setup.inf
02/13/2018  03:50 PM                4,176 DtcInstall.log
02/13/2018  03:53 PM                8,115 comsetup.log
02/13/2018  03:53 PM                22,863 diagerr.xml
02/13/2018  03:53 PM                22,863 diagwrn.xml
02/13/2018  03:53 PM    <DIR>          Registration
02/13/2018  03:53 PM    <DIR>          Tasks
02/13/2018  04:16 PM    <DIR>          SoftwareDistribution
02/13/2018  06:41 PM    <DIR>          Containers
02/13/2018  06:42 PM    <DIR>          ServiceProfiles
02/13/2018  06:44 PM    <DIR>          Help
```

When you see -- More -- *Continue pressing the space bar* to step through the directory one screen at a time until you see the prompt. If there are too many files to display, you may key Ctrl-C to quit.

### **Piping and Output Redirection**

From now on the commands will again be acting on the D drive.

Type:

**D:** <Enter>

You should see the prompt D:\Lab1\Letters\> because this was the last directory you used before you switched to the C drive.

Type:

**DIR | SORT > Lst.dat** <Enter>

The output of the DIR command is sent as input to the SORT filter. The list of files is sorted in a temporary file and sent to file *Lst.dat*. If the *Lst.dat* file does not exist it will be created; if it exists it will be overwritten.

Type:

**TREE /F** <Enter>

Do you see *Lst.dat* file in subdirectory *Letters*?

Type:

**TYPE Lst.dat** <Enter>

to see the contents of file *Lst.dat*.

If the *Lst.dat* file is too large and scrolls off the screen, type:

**TYPE Lst.dat | MORE** <Enter>

### **Copying and Erasing Files**

You can copy files to different directories or erase files from directories. The general form of a COPY command was presented earlier. It has two positional parameters for the source and the destination. Let's discuss several examples.

The command

D:\>COPY Member.3 C:

issued from the D:\> prompt would copy a file name *Member.3* from the root directory of the D drive to the current directory (we do not know what it is) of the C drive, though the last time you left the C drive, you were in the Windows subdirectory.

The command

*C:\WINDOWS>COPY John.dat D:\Lab1\Letters*

issued from C:\WINDOWS> prompt would copy a file named *John.dat* residing in the *WINDOWS* directory on the C drive to directory *Letters* on the D drive. The name of the copied file would be the same, i.e., *John.dat*. If you were to assign a new name such as *John\_Copy.dat* to the copied file, the command would look like:

```
C:\WINDOWS>COPY John.dat D:\Letters\John_Copy.dat
```

*D:\>ERASE Prog1.c*

The command issued from the D:\> prompt would delete file *Prog1.c* from the root director of the D drive.

Note that none of the above examples would actually work as you do not have *Member.3* and *Progl.c* files in the root directory of the D drive and file *John.dat* in the WINDOWS directory of the C drive.

Now let's try now several commands.

Type:

**COPY \Lab1\Letters\First\_Names.txt \Lab1\Book <Enter>**


↑                      ↑  
space                  space

Note that in the source part `\LabI\Letters\First_Names.txt` of the `COPY` command, you have typed a full path to the file `First_Names.txt`. However, the `D:\LabI\Letters>` prompt indicates that the `\LabI\Letters` subdirectory is the current directory and `D` is the current drive. As a result, the `COPY First_Names.txt \LabI\Book` should work too.

Type:

**DIR \Lab1\Book** <Enter>

You should see file *First\_Names.txt* in subdirectory Book.



The screenshot shows a Windows Command Prompt window with the following text:

```

C:\> Command Prompt
D:\Lab1\Letters>dir \Lab1\Book
Volume in drive D is KINGSTON
Volume Serial Number is D5D0-EEAF

Directory of D:\Lab1\Book

10/20/2018  12:41 PM    <DIR>          .
10/20/2018  12:41 PM    <DIR>          ..
10/20/2018   01:25 PM                29 Asgn1.txt
10/20/2018  12:59 PM                44 First_Names.txt
               2 File(s)                73 bytes
               2 Dir(s)           141,590,528 bytes free
  
```

To erase this file, type:

**ERASE** \Lab1\Book\First Names.txt <Enter>

Remember that you are in subdirectory *Letters*, so the above command provides the path to the file. If I had just typed *ERASE First\_Names.txt*, I would have gotten an error message that the file does not exist.

## **Wild cards**

Wild cards are very useful to refer to a group of files. Wild cards can be very efficiently used in the COPY and ERASE commands to copy or erase a group of files.

The asterisk "\*" represents multiple characters.  
The question mark "?" represents one character.

Examples:

The command

```
C:\>ERASE D:*. *
```

though issued from the C:\> prompt would erase all files in the current/default directory of the D drive. The name of the current directory on the D drive is not shown.

The command

```
D:\PROGRAMS>ERASE Prog?.c
```

would erase files whose names are, for example, *Prog1.c*, *Prog2.c*, *Prog9.c*, *Proga.c*, *Progz.c*, etc. The command would act on 5 character long file names. It would erase the file names whose first four characters match *Prog* and the 5<sup>th</sup> character could be any character.

The command

```
C:\WP>COPY John*.txt D:
```

would copy all files whose name start with *John* followed by anything else and have extension *txt* from the *WP* directory on the C drive to the current directory on the D drive.

Type:

```
COPY \Lab1\Letters\*.txt \Lab1\Book      <Enter>
      ↑           ↑
    space       space
```

The command will copy three files: *First\_Names.txt*, *Sally.txt*, and *Tom.txt* from subdirectory *Letters* to subdirectory *Book*. To see the 3 files copied to subdirectory *Book*, you need to display the contents of subdirectory *Book*.

Type:

```
DIR \Lab1\Book      <Enter>
```

Remember, you are in subdirectory *Letters* that is under subdirectory *Lab1* that is under the root directory \, so in the above command we had to provide the absolute/full path from the root directory to subdirectory *Book* through subdirectory *Lab1*.



You should see D:\Lab1\Letters\ prompt.

Type:

**CD ..**

<Enter>

to move to directory *Lab1* which is the parent directory of directory *Letters*.

Type:

**TREE /F > Lab1\_Tree**

<Enter>

Your *Lab1* directory structure and contents have been saved to file *Lab1\_Tree*. You will paste this file into your Lab 1 Report,

or type:

**TYPE Lab1\_Tree**

<Enter>

and use the *Alt-PrtScr* keys to capture the full screen output (full window) from command *TYPE Lab1\_Tree* and paste that window into Lab 1 Report.

**Turn in:**

- (1) Hardcopy of Lab 1 Report. The template for the report is in the Assignments\Labs on Blackboard.