



QUOTE

*PERHAPS A WRITER'S Thoughts
FROM EXPERIENCES MAY BECOME
READER'S EXPERIENCE FROM
THOUGHTS.....*

BATCH PROGRAMMING

DECODED – I

[FORNEWBIE]

By Kvc

www.thebateam.org

Index

1. 10 Lines About this Book
2. Why Batch Files???
3. Basic Requirements for Batch Scripting / Programming ...
4. Modes of interaction with System
 - i. **Interactive mode**
 - ii. **Batch Mode (Silent Mode)**
5. Introduction to CMD.exe
6. Introduction to COMMAND.com
7. Difference between CMD.exe and Command.com
8. How Cmd Works?
9. What is a Command?
10. Types of Commands :
 - I. **Internal Commands**
 - II. **External Commands**
11. How Cmd reacts to Your Commands?
12. Some Basic Commands of Cmd :
 - i. **Echo**
 - ii. **Pause**
 - iii. **Cls**
 - iv. **Title**
 - v. **Dir**
 - vi. **cd**
 - vii. **md**
 - viii. **Goto**

10 lines about book

The content in this book is written after a long experience of batch programming and definitions of various terms may vary from the original definitions as this book is purely based on Experience, not on theoretical knowledge and it takes many hours to think and type. No data is directly copied from internet, as it mayn't be correct.

I'm considering you have knowledge about various Computer programming terms Such as parameter, function, Console etc.

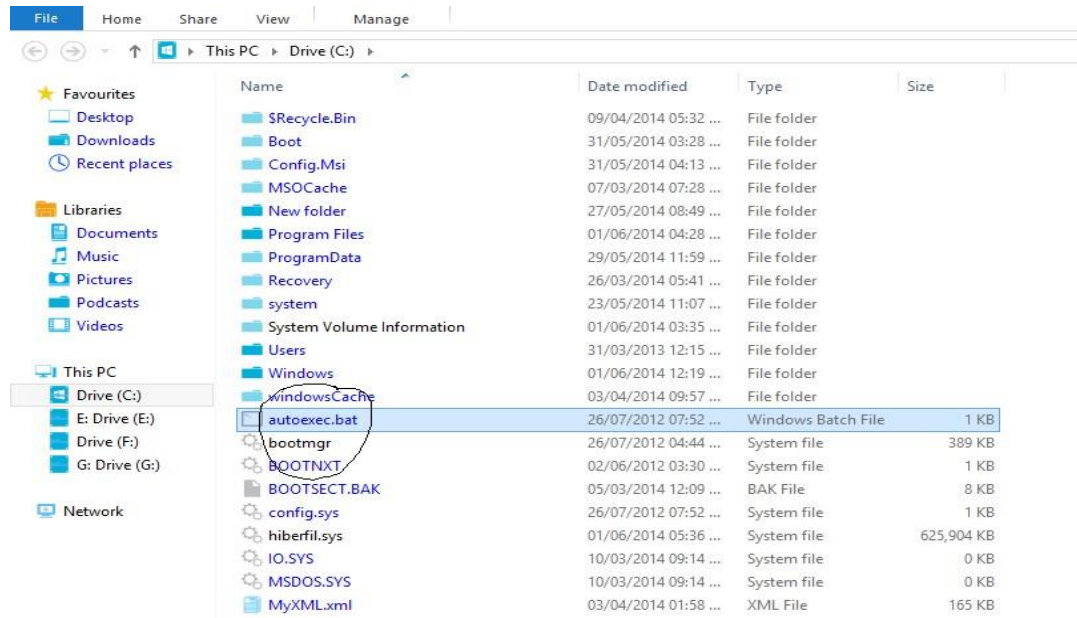
After reading **Complete Series of this book**, you will easily approach from **newbie** to **Beginner** and then to **Programmer** and then to **Advanced User** of batch programming. In this book we will take a look back to the old concepts of computer programming, and you will understand all major and minor Concepts of coding ...

Only Part-I of this Series contains too much theoretical concepts of batch programming...because, before you start batch programming – you should know these basic concepts. However, when I started batch programming I didn't know these basics....:D ,but if you understand them...you'll easily learn batch programming.

Why batch files ??

It is a nice Question to ask, there are following reasons, due to which Microsoft still using these files in their OS's:

- Automate common key-strokes (means ,no need of executing Bunch of commands again & again when OS's Startup)
- **Autoexec.bat** is the File which Runs Automatically while OS loads ... (some OS loads **autoexec.nt** , which is also a batch file)



- Single command to do many tasks... (You'll see it later ...In this Book's Series)
- Safety from dangerous DOS commands i.e. Parameters can be added
 - e.g. If you try to Delete a file via DOS ...it simply deletes it, even if you typed a wrong file name...you'll not be able to recover that file easily ...
- But in Cmd (or batch programming...), If you try to Delete a File, it Confirms twice before deleting that file...So, It increases Safety ...hence less Dangerous!!
- Reduces complexity and confusion
 - I.e. you can use any name you want for Your Own Command ...
- Fast Compilation and running of code

Requirements for batch scripting

As a newbie concept I am assuming that you are totally new to batch programming and don't know anything about it. So, I'm including this topic too... 😊

1. You Must have Windows NT based OS (**that's obvious**)
2. File must have ".bat" or ".cmd" extension.
3. Must be ASCII (not Unicode or other)
 - <3rd Point doesn't matter really, but affects your Program layout-You'll notice this later...>
4. Editors for programming...

You can use windows in-built software “notepad.exe” or any other external software like

- a. Notepad++
- b. Advance bat 2 exe
- c. WordPad
- d. WinWord etc...

Goto ‘www.thebateam.org’ to get additional tools ... #kvc 😊

- 5. Very less RAM and memory space required (depending on length of program)

Modes of interaction with system machine

- **Interactive mode.**
- **Batch mode (silent mode)**

1. Interactive mode: - when you Start cmd.Exe from your run Dialog Box, and typing a command **e.g.** a copy command, and if the Destination file name already exist there, it asks you for a selection overwrite (y or n). **This is called interactive mode,**

While in 2nd one (**Batch Mode**), if the file already exists there, it ask you for overwrite in interactive mode but here the little change in silent mode i.e. it will not ask for your (Y/N) input, it overwrite directly the file, and the syntax of command in batch mode is slightly different than syntax in Interactive mode.

Syntax: Copy “filetocopy.extension” “Destination” <In interactive mode>

Syntax: Copy /y “filetocopy.extension” “Destination” <In Batch mode>

While you may use interactive mode’s syntax in any batch file and the program pauses until you confirms the asked condition...

Introduction to Cmd (command prompt.)

It may be roughly defined as the compiler to your commands or input, which convert them to binary and then runs on system and gives you output according to that.

You can find it in the location probably in “c:\windows\system\cmd.exe”, if C: is your system drive.

Opening “cmd.exe”

Start > run > Cmd + Enter.

Intro to “command .com”

Hmmm..... if you surf through the path “C:\window\system32” in your system ,you will see many files, the file with which we are dealing with is ‘cmd.exe’ ,then why we are talking about ‘command.com’ because if you search ‘command.com’ there, you’ll find this file there too. .and this file can be defined as old version of ‘cmd.exe’ file (as per my point of view).

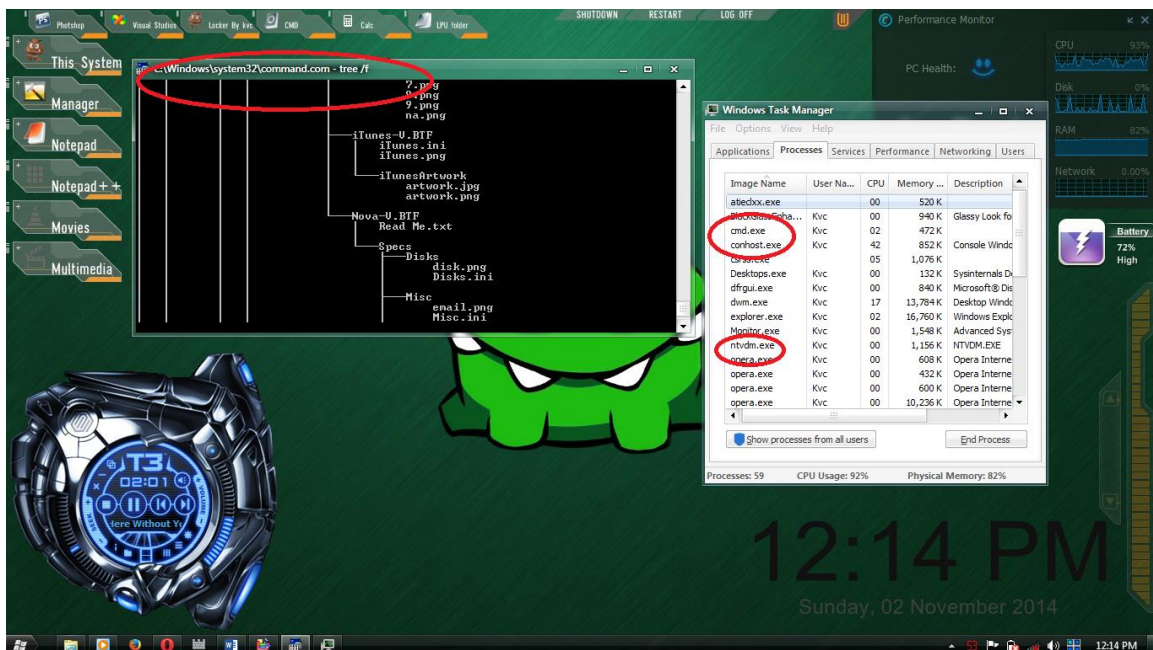
You could also run the same commands on ‘command.Com’ console by double click on it but the command execution generally takes 1 to 2 seconds (depends on system properties).

While in ‘cmd.exe’ it direct executes command, without any delay (or delay in ms).

D/f b/w ‘cmd.exe’ and ‘command.com’

Now, by reading about both ‘cmd.exe’ and ‘command .com’ you can easily understand the d/f b/w them. But I’m Specifying more technical Difference between them here...

Windows NT/Windows 2000 supply cmd.exe and command.com. CMD.EXE is the Windows NT command line interface, it’s NOT a DOS window. COMMAND.COM is a 16-bit DOS application which is used for older DOS compatibility and actually runs inside the NTVDM (NT Virtual DOS Machine) due to its 16-bit nature.



What may surprise you is the fact that COMMAND.COM and CMD.EXE have almost identical features and this is because any command entered in COMMAND.COM is packaged up and sent to CMD.EXE for execution and thanks to this COMMAND.COM can take advantage of all the functions and facilities of CMD.EXE. This is possible as the version of COMMAND.COM shipped with NT IS a special NT version designed to pass all execution to CMD.EXE.

If you had Task Manager running you would actually see a CMD.EXE process started when executing commands in COMMAND.COM.

- Now you know, why command.com takes more time to execute any command as it has to call first cmd.exe and then the command is executed.
- Command.com don't save history of typed commands as cmd Does...as you can run previous commands just by pressing **UP arrow** key and enter.
- When you run a 32-bit console program, it is executed by `cmd`; when you run a 16-bit console program, it is executed by `command`
- With **Command.com** we do not have the long name support. Typing in '*cd program files*' yields the error '*Too many parameters – files*', whereas this command works fine in **Cmd.exe**. In fact to get this to work in **Command.com**, we have to type '*cd progra~1*'.

Source: <https://windowsitpro.com>; <https://www.jameswiseman.com/blog>;

But batch scripts are running in **cmd.exe** console, so we mainly consider about **cmd.exe** in this book.

How 'cmd.exe' works?

Why we have to know this?

Before starting scripting, you should better know about the environment in which you are scripting, like in the same manner you are living on a rock, which is revolving around a star and the star is revolving about its axis in a galaxy, which is revolving around a big black-hole. So, to know about that rock (**earth**), you should better to have knowledge about its Environment, in the same manner to have Knowledge on the virtual platform of coding (scripting), you should have knowledge about its Environment.

By the way the example was too deep, **ha-ha.....** I know that.

So when we run **cmd.exe**, it shows as a **black screen** called as **console** to take our input in letter or words...

Cmd command-line works on a concept of **master-servant relationship**. Master gives the command to servant to do a job and he accomplished the task, if any mistake or problem happens, Servant directly tells the master About the Problem... Here...You are the master **(to give commands)** and Cmd serves as servant **(to work on given commands)**.

What is a Command?

A command may be defined roughly as the instructions in the form of words or Phrases to the machine to have specific type of work.

Try: **Dir C:**

The above command will show you list of all files and folders in your C: drive.

Types of command

As now we know Batch Programming is platform where we interact with the system in the form of commands... So, you should better know about the types of Commands too...

Let's Again take the example of **Master-Servant relationship**, If the Servant don't know what my master is trying to say...then he can't do what the master wants from him. So, then master have to tell servant in those words which servant can easily understands... The words known by servant are definitely from his native language...and Words that Servant don't know can be from any other language...

In same way ... the CMD has two types of Commands:-

(1) Internal commands

(2) External commands

(1) Internal commands: (also known as inbuilt commands)

Some of the commands (**basic commands**) are pre-defined in **cmd.exe** and can be used for various purposes, these are known as internal commands. Same as **servant's native Language**.

1. They can't be changed or modified by user or programmers.
2. You can't add or remove any extra parameter in these types of commands.

(2) External commands: (also known as user defined or programmer command)

This is the important part for batch scripting. You can define commands in batch programming and use them later in any of your program, your operating system do have external commands whether you define them or not, they are from Microsoft. Same as in case of Foreign language words for servant...but once he knows the meaning (**after defining by master**), then he can easily perform task according to master's command...

Microsoft puts some external commands in your system for better scripting... **e.g. Ping.exe, At.exe, Attrib.exe, Cacs.exe, Chkdsk.exe, Convert.exe** and many more...

1. These commands **can be modified** or changed by user or programmer {except Commands defined by Microsoft}
2. You can **add extra parameters** or change the name of the command as per your requirement (except for Microsoft's pre-defined external commands)

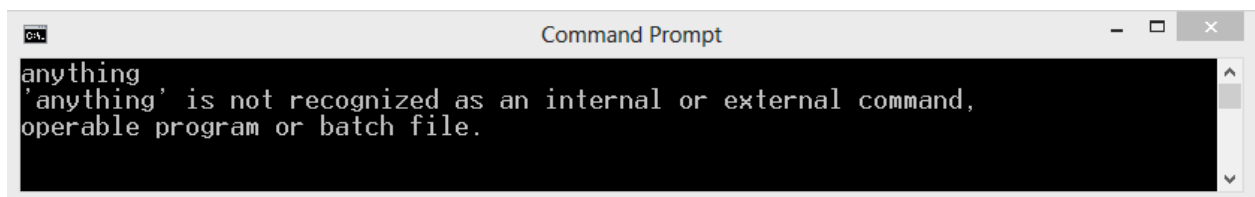
Examples:

1. **Inbuilt command**: md, cd, dir, Del etc...
2. **External commands {by Microsoft}**: Ping.exe, At.exe, Attrib.exe, diskpart.exe, defrag.exe etc... you can find them in the same directory where **cmd.exe** file exists (i.e. "C:\windows\system32")

How cmd reacts to your command?

When you write anything in CMD console (**black-screen**), it first checks that if the input is **any internal command**, if not then it checks whether it *is any external command?*

For checking external commands, it check first that, if any **‘.exe’ or ‘.com’ or ‘.bat’ or ‘.cmd’** file present in **‘C:\windows\system32’** Folder or in Current Folder... with same name as your input in console window. **E.g.** If you write **‘anything’** in Cmd Command console...



- First it checks if **Anything** is an internal command or not...
- Then it checks, if it is **an external command** or not **i.e...** it checks if any **‘anything.exe’** or **‘anything.bat’** or **‘anything.cmd’** or **‘anything.com’** file present in default command path or in current directory ...
- If cmd again fails to find any related file, then it show an Error Message on screen as **“anything is not an internal or external command.”** [AS Shown in Image]

FEW BASIC COMMANDS OF CMD

Note: CMD is not case sensitive. (Except the '*For loop*')

You can get help about any command by opening cmd console and typing '/?' after name of the command. E.g. **Echo /?**

As you are reading this book from start, and you didn't get any kind of scripting content. And you may be thinking that ... Is that guy really knows anything about **Batch Programming** OR he is just Wasting my precious time... the time in which I can go through some of my friends on **Facebook...or** the time in which I can surf through some porn data...and enjoy the Life. ;)

But believe me, **I'm not wasting your time...** Let's start from **8 Basic Commands of Batch scripting...** that you'll definitely use in any of your future batch programs ...

As they are basic commands ...So, it is natural that they are Internal Commands. ☺

1. The ECHO Command :

When you will start scripting in batch, you'll mostly use '**ECHO**' Command.

PURPOSE: This Command prints Text on cmd Console ...like "**Printf()**" in C programming.

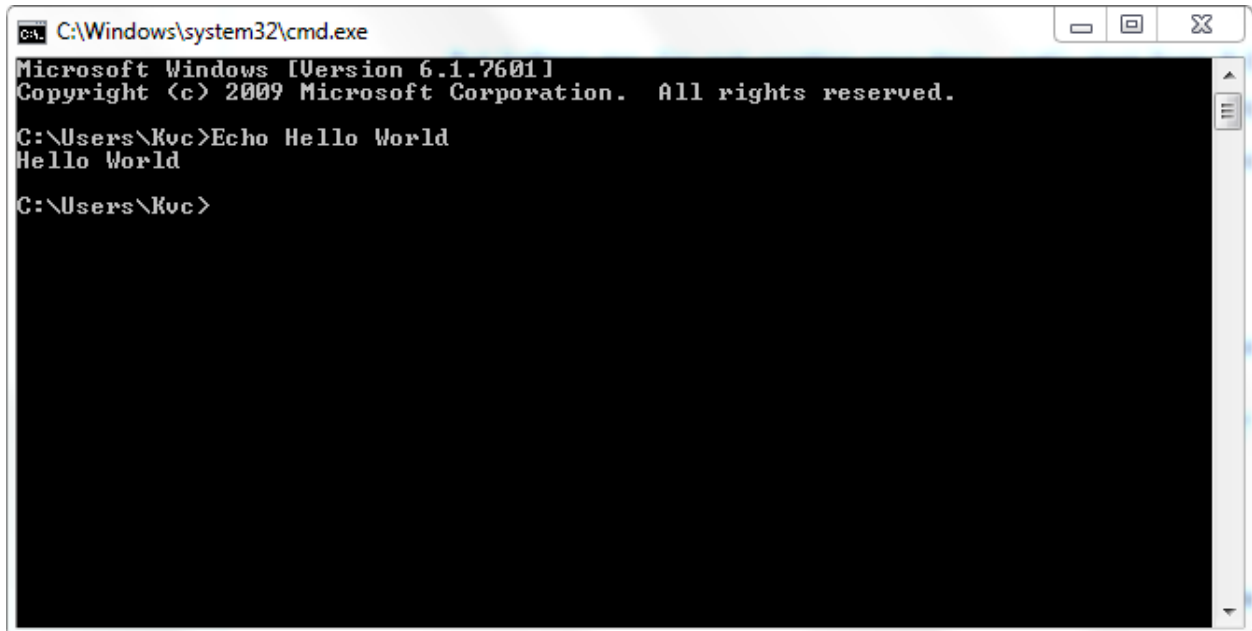
Syntax: ECHO [Text to print]

As an example is always better than a lot of words...so, let's look at the following Example -

The traditional "**hello world**" program can be written with echo command only.

Open Cmd and type following Code.

Code: **Echo hello world**

A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window content displays the following text: 'Microsoft Windows [Version 6.1.7601] Copyright (c) 2009 Microsoft Corporation. All rights reserved.' followed by the command prompt 'C:\Users\Kvc>'. The command 'Echo Hello World' has been entered, and its output 'Hello World' is displayed on the next line. The prompt 'C:\Users\Kvc>' is shown again on the following line, indicating the command has been executed.

As you will see the same type of Cmd console as in the Picture above...But there is too much mess there and you want to see only output to your Command. Then you have to know the following –

Parameter (ON\OFF) in echo commands :

As when you open cmd console, some text is already written in cmd and a line comes automatically before you give any command. That line tells you that in which Folder path you are roaming now. To remove that line from console we use 'off' keyword after 'echo' command. This stops showing that line before your input, indirectly it clears the console screen.

You should give this a try – Type '**echo**' only in cmd console and Press Enter... What Happened?? It gives you a Message as '**ECHO is off / Echo is On.**' ... That message tells you your current Echo status...i.e. you enabled the display of that extra line or not. By default Echo is ON.

To show that line again, you can use 'ON' as keyword after echo command. To show an empty line just write "echo" then a "."(Dot) without any space **i.e... Echo.**

Time for example:

Q. Write a program (WAP) to show your name, age and Gender on CMD console.

CODE:

Echo off

Echo karanveer chouhan

Echo 19 years

Echo male

Pause

Copy the Bold text above and paste it to any text editor and save it as

firstprogram.bat

Then Double click on the saved file...and there you go!!

You successfully made your first batch Program...

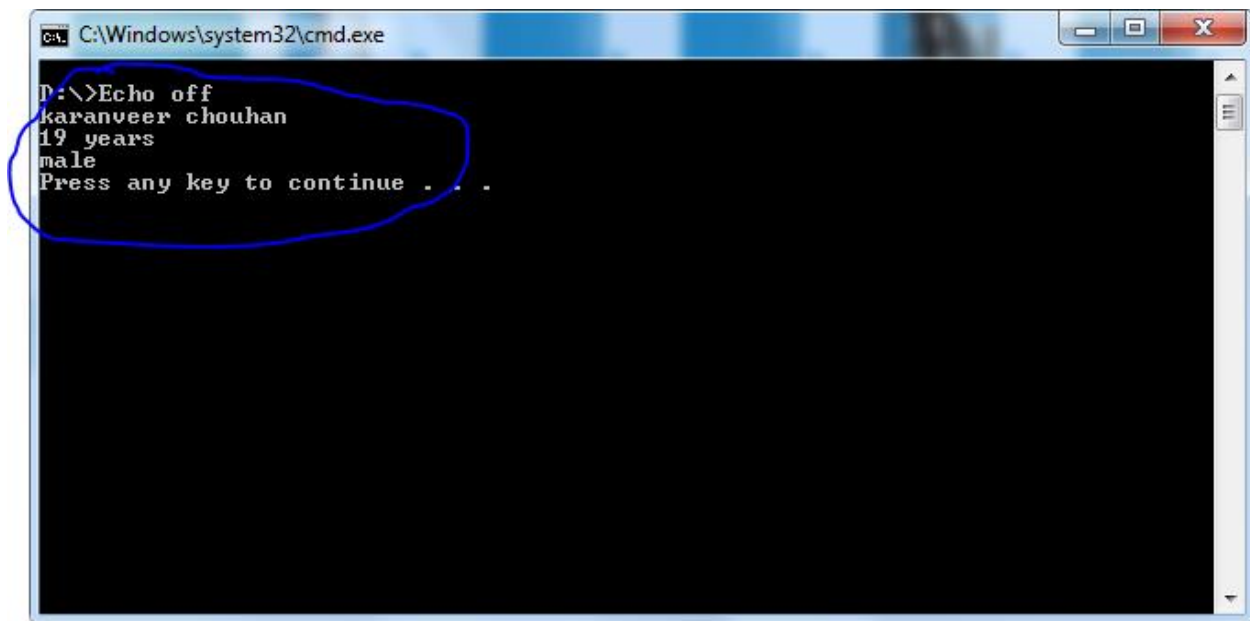
Program Explanation:

Here '**echo off**' turns off the display of extra unwanted line...

In rest of coding just put finger on 'echo' word and the Text you are able to see will be display on CMD console.

In last Line I have used 'pause' we'll talk about it in Next Section.

Output:



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with white text. The text displayed is:
D:\>Echo off
karanveer chouhan
19 years
male
Press any key to continue . . .
The first line of output, "Echo off", is circled in blue.

Most of Batch Programmers uses '@ECHO OFF' at the start of program instead of simply using 'ECHO off', because the '@' stops the display of 'echo off' (as in image above) in 1st line.

Try the same code above just by using **@echo off** instead of **ECHO OFF**.

2 The Pause Command

It is the command used to stop the batch script until any key is pressed from the keyboard.

Tip: You should use it at last of your Batch script so to see output clearly. When you use this command it will show a message “press any key to continue ...” on the console. **(as in image above)**

If you want to stop cmd console but also want, not to show the default message “press any key.....”

Just Send the Command output to NUL... where nul means nothing, and it is a default keyword.

Syntax: **pause > nul**

We'll talk about “>” operator later...

Time for example

Q. WAP to show name, age and Gender of you and your friend by using at least 2 pause commands.

=> Now, it's your choice where do you want to put the pause command or where do you want to stop the console. here I'm gonna first show my info then stopping console and then showing my Brother's info then stopping it to see the total output.

CODE:

Echo off

Echo.

Echo karanveer chauhan

Echo 19 years

Echo male

Pause > nul

Echo namish

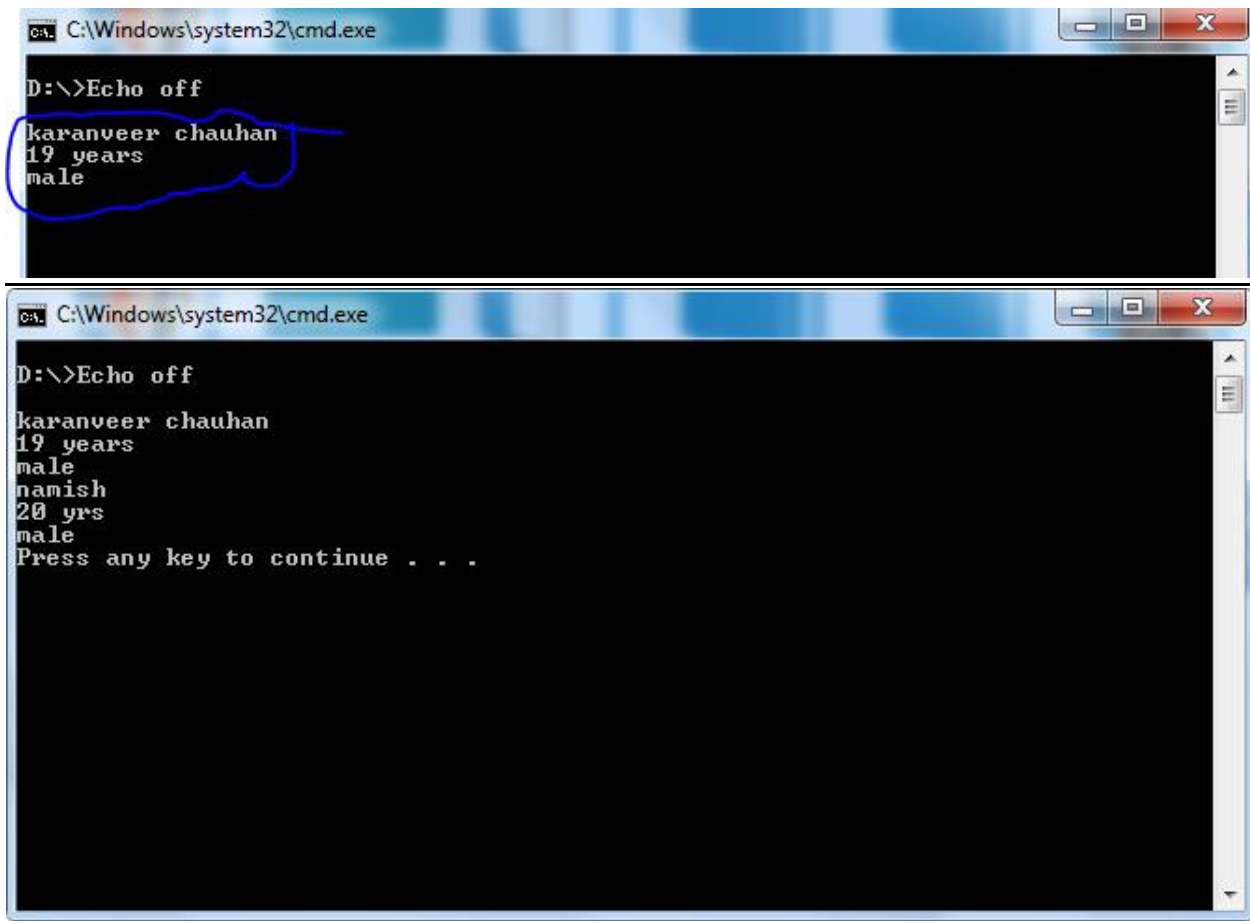
Echo 20 yrs

Echo male

Pause

Here, line 2 will show empty Line at cmd console, and 6th will be empty but when you press button, only then it will show next content according to the script and at last it stops console with message **"press any key to continue..."**

Output:



```
C:\Windows\system32\cmd.exe

D:\>Echo off
karanveer chauhan
19 years
male

C:\Windows\system32\cmd.exe

D:\>Echo off
karanveer chauhan
19 years
male
namish
20 yrs
male
Press any key to continue . . .
```

3 The Cls Command

This command is used to clear the unwanted data from the console and this Command clears the whole Console completely...

Time for example:

Q. WAP to display your name and your friends name one by one on console, but only one name should be on console at a time.

Echo off

Cls

Echo karanveer

Pause > nul

Cls

Echo namish

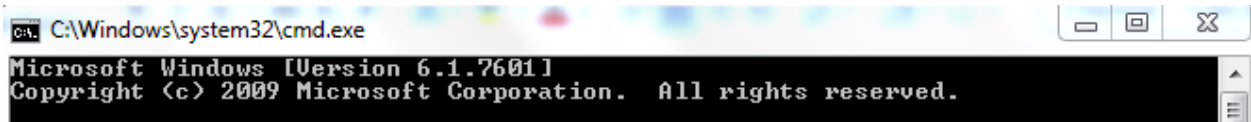
Pause

Here, after “echo off”, I’ve cleared the pre-written text by cls, and then displaying my name, and then stopping console for a key stroke.

After pressing any key then it clears Console again, and Shows ‘namish’ and again stop console with messege “press any key to continue...”

4 The Title command

This is the command used to modify text in title bar of cmd console. When you run cmd, its default title is the path in which it is open.



By using **Title** command we can change title to whatever we want.

Syntax: Title [Text for title bar]

Time for example:

Q. WAP to show use of title command in batch Programming.

Echo off

Cls

Title My first program

Echo I am learning batch programming...

Pause



Time for a break

So now you have learnt '4' commands of cmd, There is one more term in cmd called "**working directory or current directory.**"

Now "**directory**" means folder, thus the word working directory means path in system, where your cmd console is working...or applying commands.

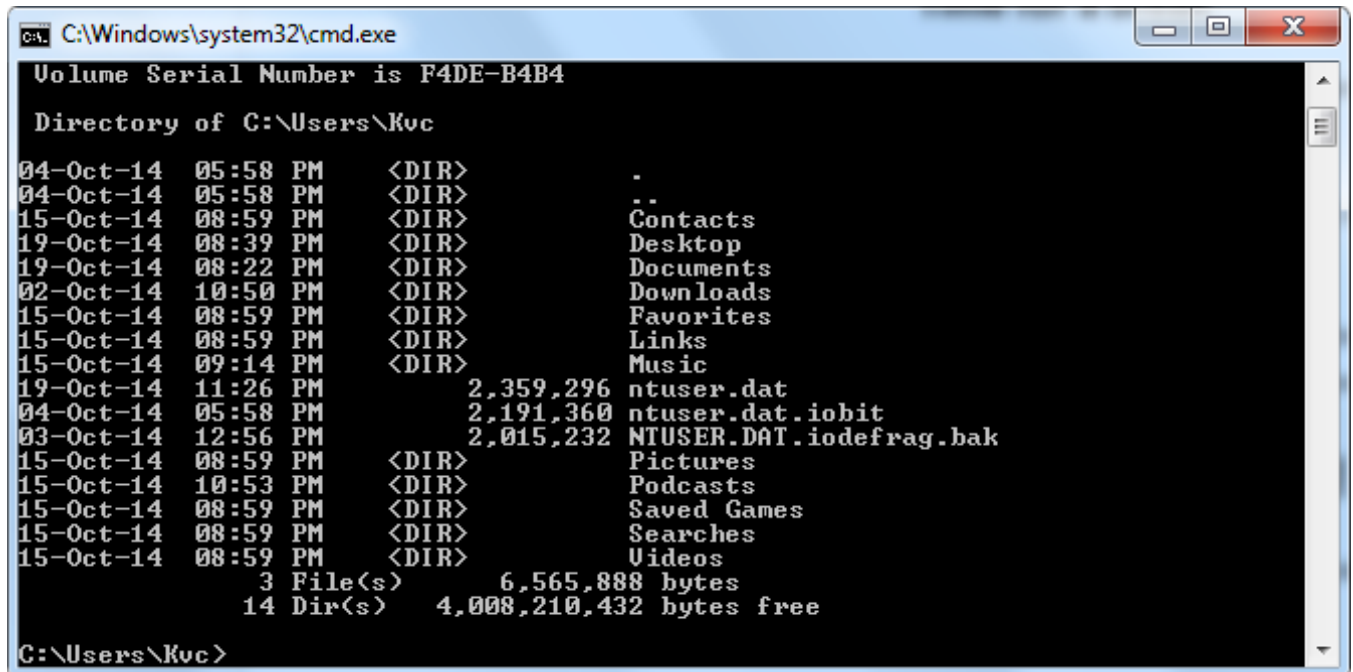
You can get this thing easily with next examples and commands, but before it you should know that, the extra line you are hiding with **echo off** was the one telling you the working directory.

E.g: **C:\Users>**

The code above denotes Working directory!

5 The DIR command

This is the command telling you about **files/folders** present in the working directory.



```
C:\Windows\system32\cmd.exe
Volume Serial Number is F4DE-B4B4

Directory of C:\Users\Rvc

04-Oct-14  05:58 PM    <DIR>          .
04-Oct-14  05:58 PM    <DIR>          ..
15-Oct-14  08:59 PM    <DIR>          Contacts
19-Oct-14  08:39 PM    <DIR>          Desktop
19-Oct-14  08:22 PM    <DIR>          Documents
02-Oct-14  10:50 PM    <DIR>          Downloads
15-Oct-14  08:59 PM    <DIR>          Favorites
15-Oct-14  08:59 PM    <DIR>          Links
15-Oct-14  09:14 PM    <DIR>          Music
19-Oct-14  11:26 PM             2,359,296 ntuser.dat
04-Oct-14  05:58 PM             2,191,360 ntuser.dat.iobit
03-Oct-14  12:56 PM             2,015,232 NTUSER.DAT.iodefrag.bak
15-Oct-14  08:59 PM    <DIR>          Pictures
15-Oct-14  10:53 PM    <DIR>          Podcasts
15-Oct-14  08:59 PM    <DIR>          Saved Games
15-Oct-14  08:59 PM    <DIR>          Searches
15-Oct-14  08:59 PM    <DIR>          Videos
               3 File(s)              6,565,888 bytes
              14 Dir(s)          4,008,210,432 bytes free

C:\Users\Rvc>
```

Type 'Dir' in cmd console and press Enter...you'll see output something like above image...
The '<DIR>' is telling that Corresponding item in list is a folder.

Syntax: DIR [drive:][path]

Some Important Switches/Parameters of Dir command

As you are still newbie, and 'dir' is very advanced command, so I'll tell you only some important parameters of this command... If you want to know about all the parameters...just Open cmd and type '**Dir /?**'

Switch working

/A	Displays files with specified attributes.	
Attributes:	D =Directories	R =Read-only files
	H =Hidden files	A =Files ready for archiving
	S =System files	I =Not content indexed files
	L =Reparse Points	- (minus) Prefix meaning not
/B	Uses bare format (no heading information or summary).	
/P	Pauses after each screenful of information.	
/Q	Display the owner of the file.	
/S	Displays files in specified directory and all subdirectories.	

Try Using 'dir' command with these parameters and the combination of different parameters together...

TIP: Try using /s and /b switches (or parameters) together to get full path of any file.

6 The CD command

CD stands for '**Change directory**', it is newer form of older 'CHDIR' command which is same command as 'CD', now from the command name, you can easily predict that this is the command used to change working directory of cmd console from one folder to other folder...

Syntax: CD [/d] [Folder name or Drive:]

Where, /d : To enter into different drive...

[..] : To change working directory up one level to parent folder...

Usage:

Cd .. // to go up one level (previous folder)

OR

Cd [folder name] // to change directory to specified folder

OR

Cd /d [drive letter:] // to roam into different drive

TIP: Open cmd, and type any Drive letter (E.g. 'D:', if D: is available...), what happened?

The working directory changed to that drive...without the use of **CD command**. But this trick is not valid for full path...so, you have to use CD command in that case...

7 The MD command

This is the command used to create a new folder in Working Directory (folder).

Syntax: md [folder name]

E.g: md folder1

but if the folder-name contains spaces, such as “New Folder”, then Use ‘Double Quotes’ around the name of Folder as follows, otherwise cmd will created two or more folders(depending upon no. of spaces) all having names as parts of folder name separated by spaces...

e.g.: **md** “*New Folder*”

Will create only one folder named ‘New Folder’

While, **md** *New Folder*

Results in creating two folders named as, *New* and other one is *Folder*.

8 The goto Command

As the name of command suggests, this command changes/Breaks the direction of flow of cmd code compilation, which is originally from **top to bottom**. GOTO command only takes 1 parameter...i.e. name of the label...

Here comes the new term...'**Label**' now you'll ask, **what is a Label?**

A label may be a checkpoint inside a batch file, which can be given a name, and there is always a single colon (:) before name of any label... an example may be a better explanation for you all...

Time for an example:

Q: WAP to make a folder named 'a' and another folder 'b' inside it...and use goto statement for this purpose.

Echo off

Cls

Title learning batch...

Goto second

:first

Md b

Pause

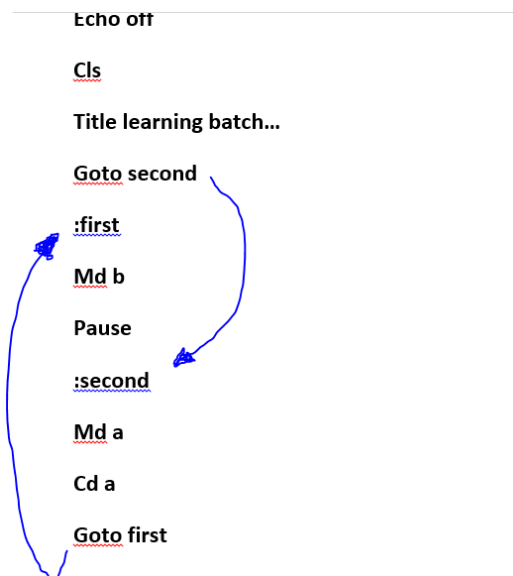
exit

:second

Md a

Cd a

Goto first



Now copy the above code and save as **'Test.bat'** ... in your Desktop...And you'll see Changes according to the following explanation.

Explanation: (line by line) – what each command is doing in the example?

Stopping display of extra mess on console...using line1

Clearing screen...for any older unwanted content

Changing title of cmd console to what we desired.

Switching the control of batch program to label named 'second'

UNDER LABEL SECOND...

Creating folder

Changing working directory...

Switching the control of batch program to label named 'first'

Creating folder

Pausing console for a keystroke

Exiting program

Finally This Part Ends

After reading contents of this Book, I think that you successfully understood the basics of Batch Programming... What we had done till now, is making of static programs...that will always show your name, age etc. on console.

If you like the Way I've explained...do tell me because your one message can boost me up for writing more books and programs in batch for you all.

Check out Next part of this series for a deeper understanding of Variable handling in Batch programming, and to make Dynamic programs in batch Programming.

Blog: www.thebateam.org

#kvc