# 6 BATCH COMMANDS IMPLEMENTATION

Below is a listing of commands used in a batch file with additional information about each of the commands.

**Tip:** Just like all commands, all batch file commands are not case sensitive. However, we listed the batch file commands in all caps to help with identification.

**@**

The at symbol does not echo back text after the symbol. The @ is most often used as **@ECHO OFF** to prevent any of the commands in the batch file from being displayed, just the information outputted by the command.

---

**%1**

The percent followed by a numeric value, beginning with one, allows you to add matched variables to a batch file. The line below is an example of what can be used in a batch file.

ECHO Hello %1

With a batch file containing the above line if you type myname (name of bat file) and then your name, as shown below.

myname Bob

It would output "Hello Bob" because "Bob" is the first matched text.

**Tip:** You can keep going to %2, %3, and so on. For example, you could use %2 for a middle name and %3 as the last name.

---

**::**

Two colons in front of any line are one of two ways of adding remarks into the batch file without displaying or executing that line when the batch file is run. Unlike REM, this line is not shown regardless if ECHO off is in the batch file.

---

**:LABEL**

By adding a colon in front of a word, such as LABEL, you create a category, more commonly known as a label. A label allows you to skip to certain sections of a batch file such as the end of the batch file. Also see GOTO.

**CALL**

A call is used to run another batch file within a batch file. When the batch file that is called is completed, the remainder of the original batch file is completed. If the batch file does not exist, you get an error.

**CHOICE and SET**

See our how to use choice and set in a batch file page for an example of how you can create options in your batch file. Further information about each of these commands can also be found on the choice and set page.

**CLS**

Just like the DOS command would clear your screen. We find it helpful to run the cls command at the top of your batch file to clear any previous commands or output and make any output from the batch file easier to find and read.

**ECHO**

Echo a message in the batch file. Such as **ECHO Hello World** prints *Hello World* on the screen when executed.

**Note:** Without **@ECHO OFF** at the beginning of the batch file you'll also get "ECHO Hello World" and "Hello World."

**Tip:** If you'd just like to create a blank line, type **ECHO.** adding the period at the end creates an empty line.

**EXIT**

Exits out of the DOS window if the batch file is running from Windows. See the exit command page for further information on this command.

**GOTO**

Jumps to a label or section of a batch file. The goto can make it easy to jump back to the start or end of a batch file if a condition is met, or an error occurs. See our how to use choice and set in a batch file page for an example of how goto can be used.

**IF**

Used to check for a certain condition if the condition exists. If that condition exists it performs that function. See the if command for further information on this command.

**PAUSE**

Prompt the user to press any key to continue.

**REM**

One of two ways of adding remarks into the batch file without displaying or executing that line when the batch file is run.

**SHIFT**

The shift command changes the position of replaceable parameters in a batch program. See the shift page for further information on this command.

**START**

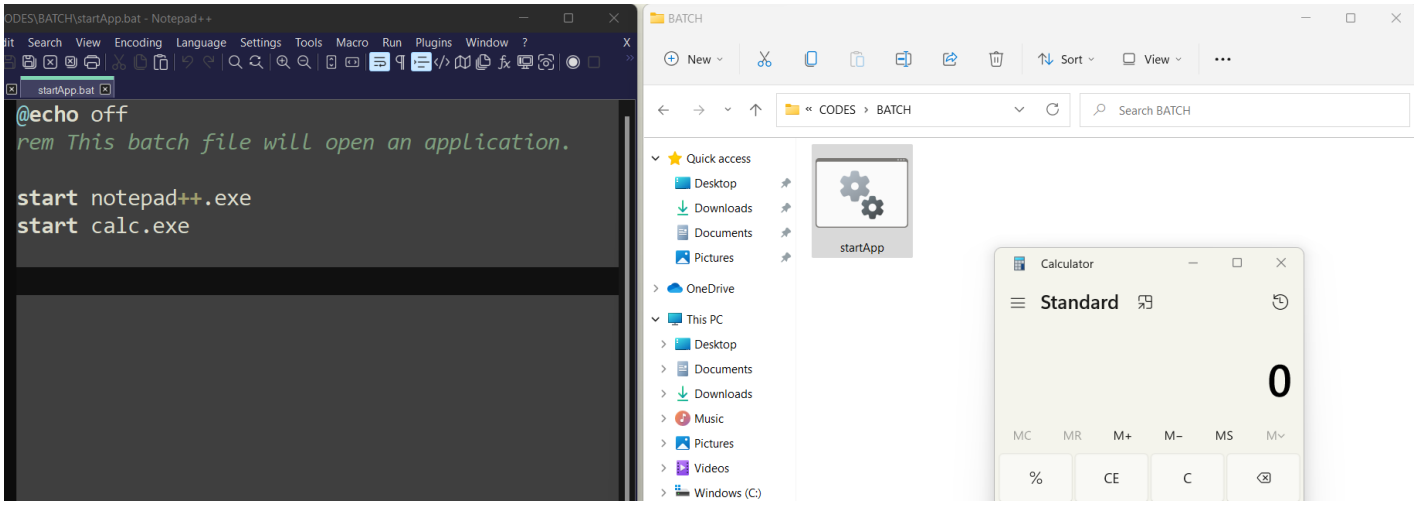
Used to open Windows programs. For example, **START C:\WINDOW\CALC** would run the Windows Calculator. The start command can also be used to start any file Windows recognizes. For example, you could start a movie or audio file in a batch file to start your default player for that file.

---

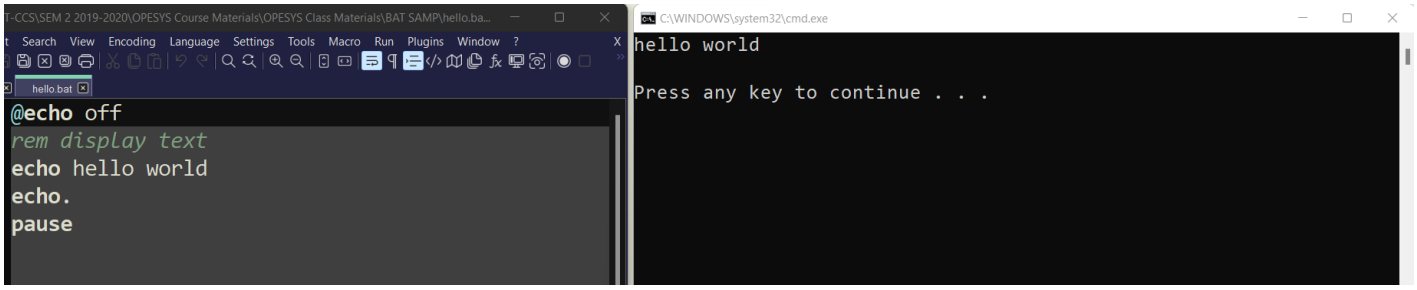
**NOTE:** To write the batch scripts, you may use any text editor. To create a batch file, you have to save your batch scripts into .bat

**SAMPLE SCRIPTS with OUTPUT**

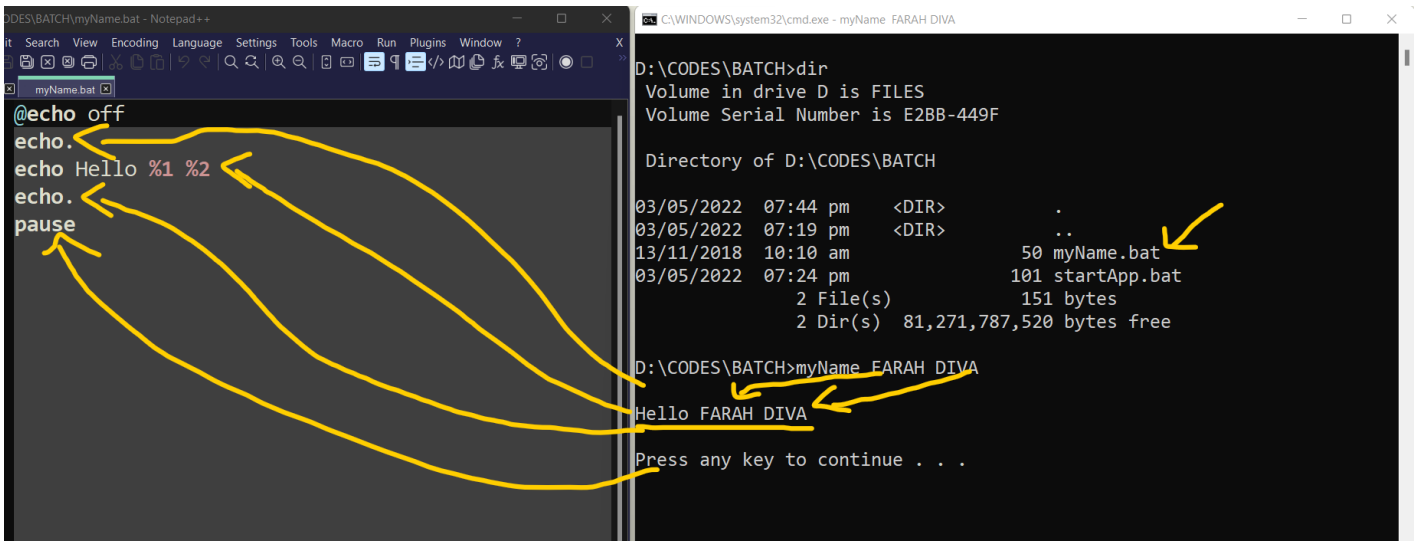
To execute this script, just double-clicked the batch file.



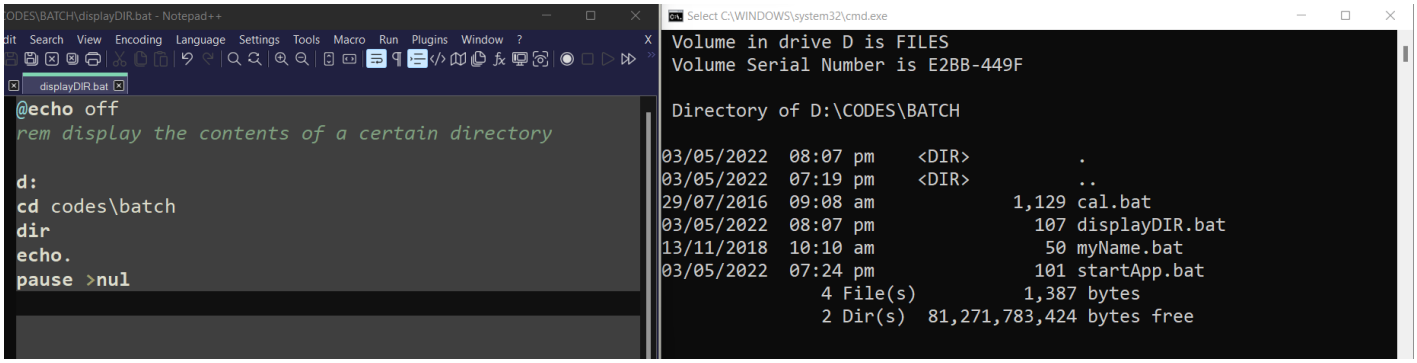

To execute this script, just double-clicked the batch file.



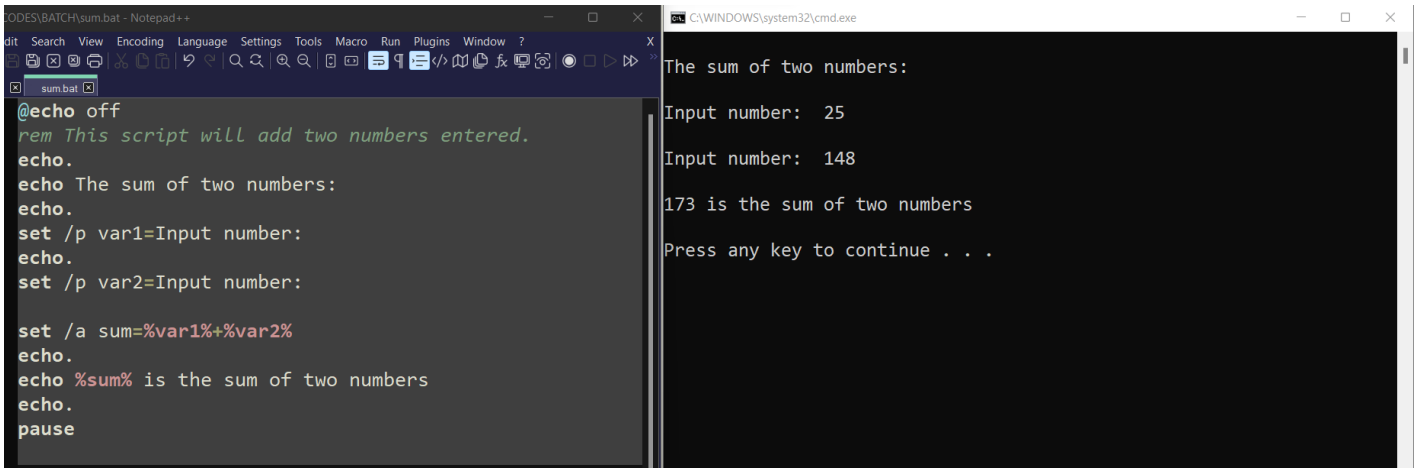
This batch file has to run on the cmd window because there is a need to input values for the generic variables on the scripts.




This batch file shows that you can make use of Windows Internal commands combine with the batch commands.




This batch file computes the sum of two numbers inputted by the user.




**Hi I'm Flashee!**

You have reached the end of the lesson. Be sure to answer the corresponding **activity of this lesson** on the activities folder of our class materials in the file server.