Windows Command Line

A 2-Hour Lab Session for First-Time Users

Updated August, 2026

Contents

1 Lab Overview

Objectives

By the end of this session, you will be able to:

- Open and navigate the Windows Command Prompt (`cmd.exe`).
- Create, copy, move, rename, and delete files and folders from the command line.
- Redirect and filter command output using `>`, `>>`, and `|`.
- Retrieve basic system information using built-in commands.

- Combine several commands to build a short text report, entirely from the keyboard, without touching File Explorer.

Prerequisites

A computer running Windows 10 or 11. No prior command-line experience is required.

Timing

Part	Topic	Duration
1	Warm-up: opening and reading the prompt	20 min
2	File and folder management	30 min
3	Text output and redirection	30 min
4	System exploration	20 min
5	Challenge exercise (report builder)	20 min
	Total	**120 min**

2 Getting Started with the Command Prompt

2.1 Opening a Command Prompt

1. Press **Win + R** to open the Run dialog, type `cmd`, and press **Enter**.
   – *or* – click the **Start** menu, type "Command Prompt", and press **Enter**.
2. A black window opens with a line similar to:

```
Microsoft Windows [Version 10.0.26100.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>
```

Tip

This document uses `cmd.exe`, the classic Command Prompt. Windows also ships **PowerShell** and the newer **Windows Terminal** app, which can host both. The commands in this lab work in a plain Command Prompt window; most also work unchanged inside PowerShell.

2.2 Reading the Prompt

`C:\Users\student>` tells you two things: the current **drive** (`C:`) and the current **folder** (`\Users\student`). Everything you type is interpreted relative to this location unless you give a full path.

2.3 A Few Habits Worth Building Early

Tip

- Press **Tab** while typing a file or folder name to auto-complete it – press it again to cycle through matches.
- Press the **Up** and **Down** arrow keys to scroll through previously typed commands.
- Press **F7** to open a pop-up list of your command history.
- Right-click inside the window to paste text copied from elsewhere.
- Commands and folder names are **not** case-sensitive in `cmd.exe`.

3 Command Reference

The tables below group the commands you will use in this lab by purpose. Keep this section open as a reference while you work through Section 4.

3.1 Getting Help

Command	What it does
`help`	Lists every built-in command with a one-line description.
`<command> /?`	Shows detailed usage and options for a specific command, e.g. `copy /?`.
`cls`	Clears the screen (does not affect anything you have created).
`exit`	Closes the Command Prompt window.
`ver`	Displays the Windows version currently running.

3.2 Navigating the File System

Command	What it does
`cd (or chdir)`	Displays the current directory when typed alone.
`cd <path>`	Changes to the specified directory, e.g. `cd Documents`.
`cd ..`	Moves up one directory level (to the parent folder).
`cd \`	Jumps straight to the root of the current drive.
`<letter>:`	Switches to a different drive, e.g. `D:`.
`dir`	Lists the files and folders in the current directory.
`dir /w`	Lists contents in a compact, multi-column format.
`dir /a`	Lists all entries, including hidden and system files.
`dir /s`	Lists contents of the current directory *and* every subdirectory.
`tree`	Displays the folder structure as a visual tree.
`tree /f`	Same as `tree`, but also lists the files in each folder.

3.3 Managing Files and Directories

Command	What it does
`mkdir (or md)`	Creates a new directory, e.g. `mkdir Reports`.
`rmdir (or rd)`	Removes an *empty* directory.
`rmdir /s /q`	Removes a directory and everything inside it, without asking for confirmation.
`copy <src> <dst>`	Copies one or more files to a new location.
`xcopy <src> <dst> /e`	Copies whole directory trees, including empty subfolders.
`move <src> <dst>`	Moves a file or folder to a new location (can also rename it).

Command	What it does
`ren` (or `rename`)	Renames a file or folder, e.g. `ren old.txt new.txt`.
`del` (or `erase`)	Deletes one or more files.
`type <file>`	Displays the full contents of a text file in the console.

Caution

`del` and `rmdir /s /q` do **not** send files to the Recycle Bin – deletion from the command line is immediate and permanent. Double-check the path before pressing Enter.

3.4 Viewing and Redirecting Output

Command	What it does
`echo <text>`	Prints <text> to the screen.
`echo.`	Prints a blank line.
`<cmd> > file`	Sends a command's output to `file`, **overwriting** it if it exists.
`<cmd> >> file`	Sends a command's output to `file`, **appending** to the end.
`<cmd1> | <cmd2>`	Pipes the output of `cmd1` into `cmd2` as its input.
`more`	Displays piped output one screen at a time (press `Space` to continue).
`findstr "<text>"`	Searches input for lines containing <text>. Add `/i` to ignore case.
`sort`	Sorts input lines alphabetically.

3.5 System Information

Command	What it does
`systeminfo`	Prints a detailed report: OS version, memory, boot time, and more (takes a few seconds).
`hostname`	Displays the computer's network name.
`whoami`	Displays the username you are currently logged in as.
`tasklist`	Lists every process currently running.
`taskkill /IM <name> /F`	Forcibly closes a running program by its process name, e.g. `notepad.exe`.

3.6 Environment Variables and Miscellaneous

Command	What it does
`set`	Displays all current environment variables.

Command	What it does
`set <name>`	Displays the value of one variable, e.g. `set PATH`.
`%date%  %time%`	Placeholders that expand to the current date/time – useful inside `echo`.
`start <program>`	Opens a program or file, e.g. `start notepad`.
`clip`	Copies piped text to the clipboard, e.g. `dir | clip`.

4 Lab Assignment

Work through the parts below **in order**. Each part builds on the files and folders created in the previous one, so try not to skip ahead. Where a task asks you to record something, write it directly on your answer sheet (or in a text file you keep open in Notepad) – you will need these notes for the final submission.

4.1 Part 1 – Warm-up (20 minutes)

Task 1.1 – Open and identify

1. Open a Command Prompt window as described in Section 2.1.
2. Run `ver` and record the Windows version shown.
3. Run `whoami` and record the username.
4. Run `cd` on its own and record the folder it prints.

Task 1.2 – Move around

1. From your starting folder, run `cd %USERPROFILE%\Desktop` to jump to your Desktop.
2. Run `dir` and note how many items are listed.
3. Run `cd ..` once. Which folder are you in now?
4. Run `cd \` and then `dir`. What do you see at the root of the `C:` drive?
5. Return to your Desktop folder before continuing.

4.2 Part 2 – File and Folder Management (30 minutes)

Task 2.1 – Build a folder structure

From your Desktop folder, create the following structure using only `mkdir` and `cd`:

```
CLI_Lab
  |-- Documents
  |-- Backup
  \-- Temp
```

Hint: create `CLI_Lab` first, `cd` into it, then create the three subfolders in a single line: `mkdir Documents Backup Temp`.

Task 2.2 – Create some files

1. Move into `CLI_Lab\Documents`.
2. Create three text files using `echo`, each with a short sentence inside, for example:

```
echo First lab file > notes1.txt
echo Second lab file > notes2.txt
```

```
echo Third lab file > notes3.txt
```

3. Run `dir` to confirm all three files exist.

Task 2.3 – Copy, rename, move, delete

1. **Copy** `notes1.txt` into the `Backup` folder (use a relative path such as `..\Backup`).
2. **Rename** `notes2.txt` to `notes2_old.txt`.
3. **Move** `notes3.txt` into the `Temp` folder.
4. From the `CLI_Lab` folder, run `tree /f` and record the output – it should show all three subfolders and where each file ended up.
5. **Delete** the `Temp` folder and everything inside it with a single command.
6. Run `tree /f` again to confirm `Temp` is gone.

4.3 Part 3 – Text Output and Redirection (30 minutes)

Task 3.1 – Write to a file with echo

From the `CLI_Lab` folder:

1. Create `readme.txt` containing one line of your choice:

```
echo This is my first CLI lab file. > readme.txt
```

2. **Append** a second line without erasing the first:

```
echo Completed on %date% >> readme.txt
```

3. Display the file's contents with `type readme.txt` and confirm both lines are present.

Task 3.2 – Redirect a listing and filter it

1. Save a full listing of `Documents` to a file:

```
cd Documents
dir > filelist.txt
```

2. Use `type` together with a pipe to show only the lines that mention `.txt`:

```
type filelist.txt | findstr ".txt"
```

3. Pipe the same file through `sort` and note whether the order of the lines changes.
4. *Optional:* pipe the output of `dir /s` on your whole `CLI_Lab` folder into `clip`, then paste it into a Notepad window to see what was copied.

4.4 Part 4 – System Exploration (20 minutes)

Task 4.1 – Explore your system

1. From `CLI_Lab`, run `systeminfo > sysinfo.txt` (this can take up to 20 seconds – wait for the prompt to return).
2. Use `findstr` to pull out two specific lines without opening the whole file:

```
type sysinfo.txt | findstr /i "OS Name"
type sysinfo.txt | findstr /i "Total Physical Memory"
```

3. Record both values on your answer sheet.

Task 4.2 – Manage a running process

1. Open Notepad from the command line: `start notepad`.
2. Confirm it is running: `tasklist | findstr /i notepad`.
3. Close it from the command line: `taskkill /IM notepad.exe /F`.
4. Run `tasklist | findstr /i notepad` again – it should now return nothing.

4.5 Part 5 – Challenge Exercise: Build a Report (20 minutes)

Task 5.1 – Combine everything

Using only commands from this lab, build a small personal report **without touching File Explorer or Notepad's mouse-driven menus**:

1. Create a new folder named `Report_<yourname>` inside `CLI_Lab`.
2. Inside it, build a file named `report.txt` that contains, in order:
   - your name and today's date,
   - your computer's hostname (`hostname`),
   - the Windows version (`ver`),
   - a full listing of the `CLI_Lab` folder (`dir`).

   Use `echo` with `>` for the first line and `>>` for every line after that, so nothing you already wrote gets overwritten.
3. Once finished, display the whole file with `type report.txt` and check that all four pieces of information are present and correctly ordered.

Tip

Commands like `hostname` print only their own output – they do not know they are being redirected into the middle of a file. Redirecting each command's result with `>>` one line at a time, in the order you want them to appear, is exactly how simple text reports are built from the command line.

Task 5.2 – Bonus: your first batch file

Everything you typed by hand in Task 5.1 can be saved as a single script and reused.

1. Open Notepad: `start notepad`.
2. Type the same sequence of commands you used above, one per line – for example:

```
@echo off
echo Student report > report2.txt
echo Hostname: %computername% >> report2.txt
ver >> report2.txt
echo Generated on %date% at %time% >> report2.txt
```

3. Save the file as `build_report.bat` *inside* your `Report_<yourname>` folder (choose "All Files" as the type in Notepad's Save dialog so it keeps the `.bat` extension).
4. Back in the Command Prompt, run it by typing its name: `build_report.bat`.
5. Confirm `report2.txt` was created with `type report2.txt`.

5 Submission Instructions

At the end of the session, submit the following:

1. Your completed answer sheet from Parts 1 and 4 (Windows version, username, OS name, memory size).

2. The final contents of `report.txt` from Task 5.1.
3. If you completed the bonus task, the contents of `report2.txt` and your `build_report.bat` file.
4. A screenshot of your final `tree /f` output from Task 2.3, showing your complete `CLI_Lab` folder structure.

Caution

Do not delete your `CLI_Lab` folder after the lab – some instructors spot- check the live folder structure in addition to the submitted files.

6 Appendix: One-Page Command Cheat Sheet

Cmd	Use	Cmd	Use
`cd`	show/change folder	`dir`	list contents
`cd ..`	go up one level	`tree /f`	show folder tree
`mkdir`	create folder	`rmdir /s /q`	delete folder
`copy`	copy a file	`move`	move/rename
`ren`	rename	`del`	delete a file
`type`	show file contents	`echo`	print / write text
`>`	redirect (overwrite)	`>>`	redirect (append)
`|`	pipe to next command	`findstr`	search text
`sort`	sort lines	`cls`	clear screen
`ver`	Windows version	`whoami`	current user
`hostname`	computer name	`systeminfo`	full system report
`tasklist`	list processes	`taskkill /IM`	close a process
`set`	show environment vars	`start`	open a program/file