**Email:** meghanabadgujar07@gmail.com

**Name:** Meghana Samadhan Badgujar  
**Batch Name:** WiproNGA\_DWS\_B5\_25VID2550

**User ID:** 34737

**Batch ID:** 25VID2550

**Date:** 9th August 2025

**Assignment Topic:**

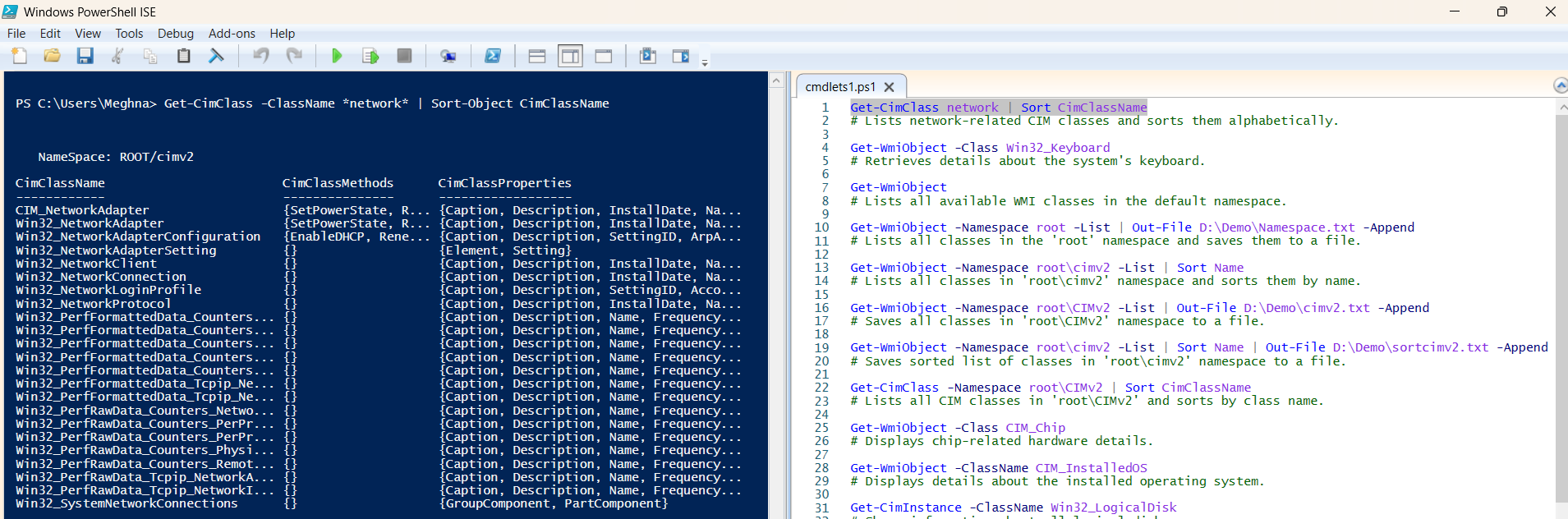
* **PowerShell cmdlets commands**

**1. WMI & CIM Cmdlets – System Information**

**Purpose: Retrieve hardware, OS, and class information**.

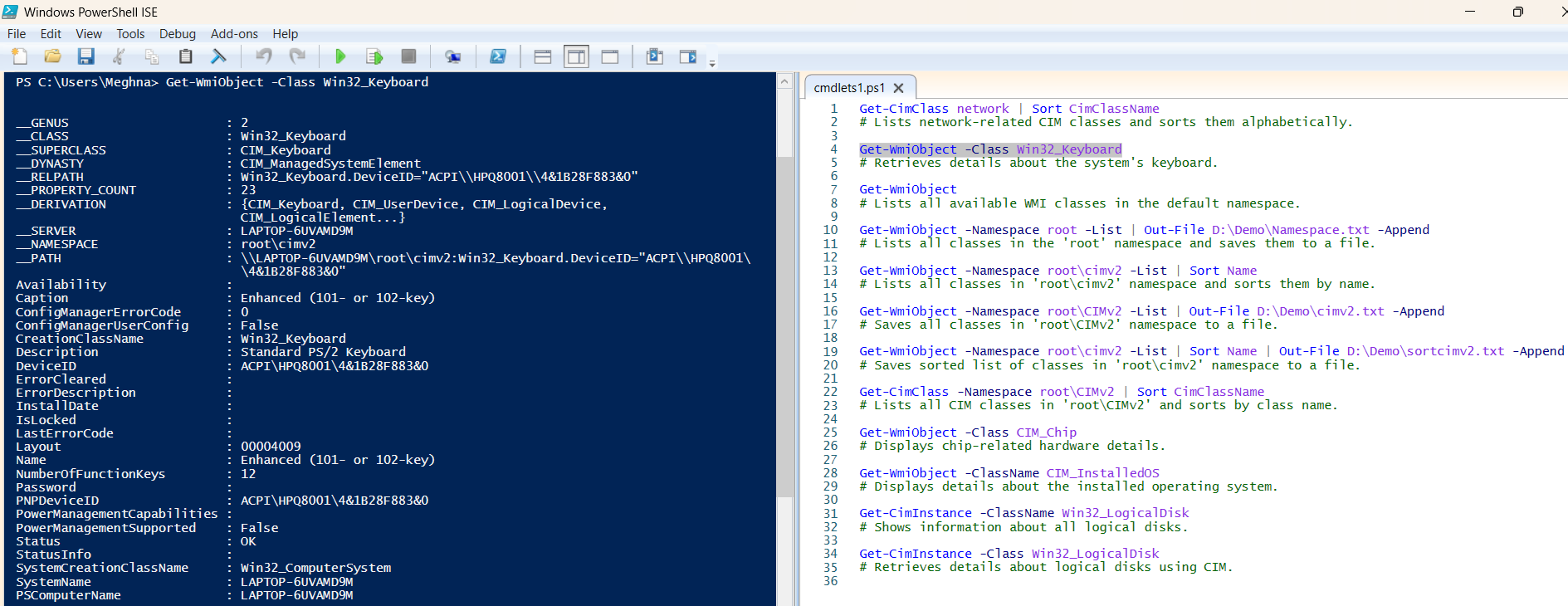
1. Get-CimClass network | Sort CimClassName

# Lists network-related CIM classes and sorts them alphabetically.



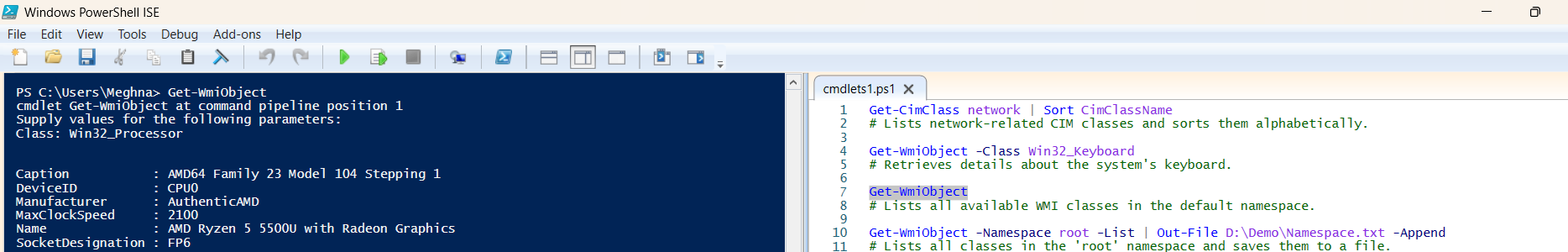
1. Get-WmiObject -Class Win32\_Keyboard

# Retrieves details about the system's keyboard.



1. Get-WmiObject

# Lists all available WMI classes in the default namespace.

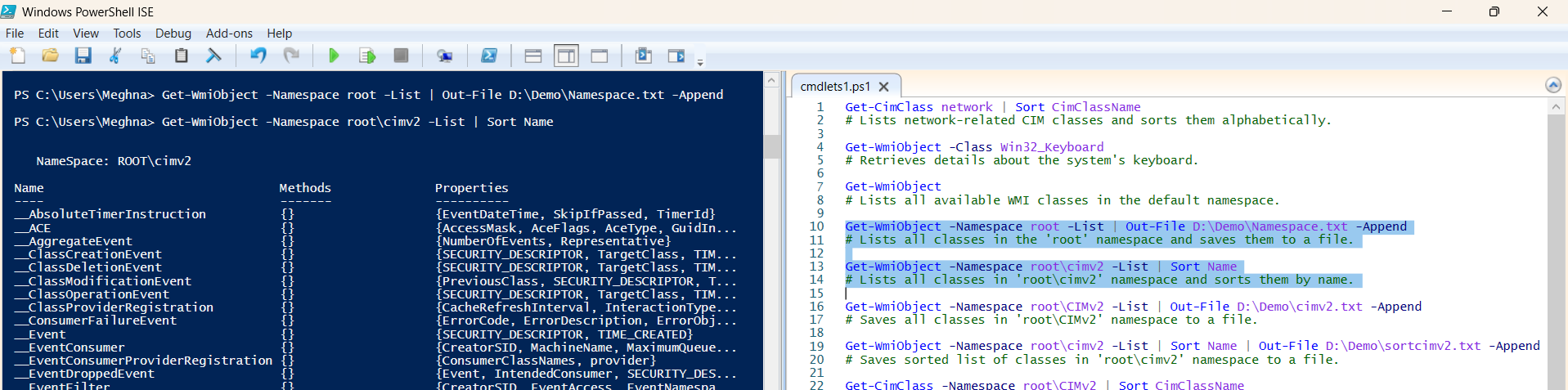


1. Get-WmiObject -Namespace root -List | Out-File D:\Demo\Namespace.txt -Append

# Lists all classes in the 'root' namespace and saves them to a file.

1. Get-WmiObject -Namespace root\cimv2 -List | Sort Name

# Lists all classes in 'root\cimv2' namespace and sorts them by name.



1. Get-WmiObject -Namespace root\CIMv2 -List | Out-File D:\Demo\cimv2.txt -Append

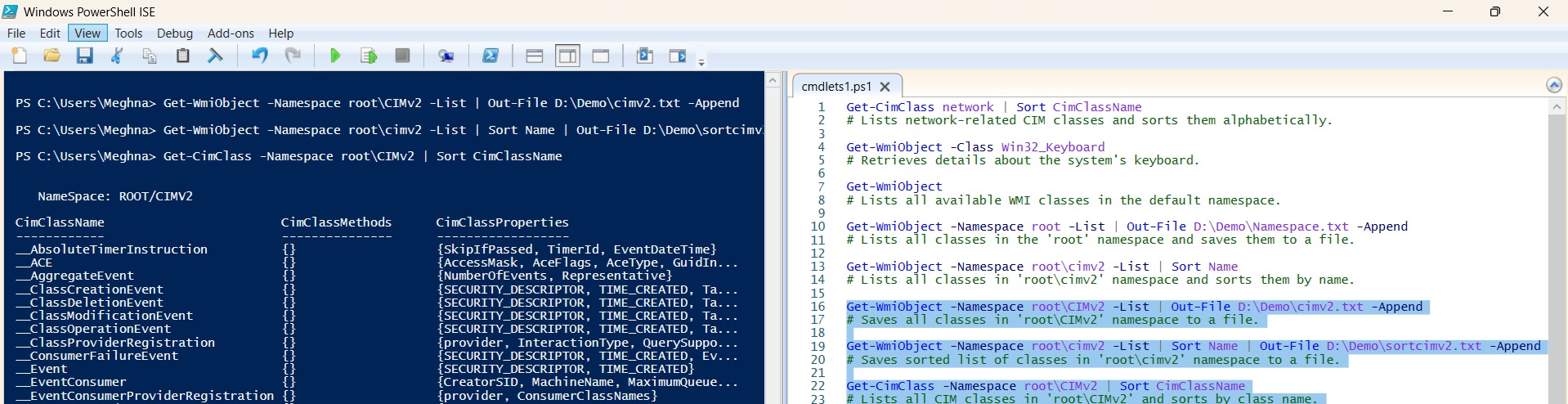
# Saves all classes in 'root\CIMv2' namespace to a file.

1. Get-WmiObject -Namespace root\cimv2 -List | Sort Name | Out-File D:\Demo\sortcimv2.txt -Append

# Saves sorted list of classes in 'root\cimv2' namespace to a file.

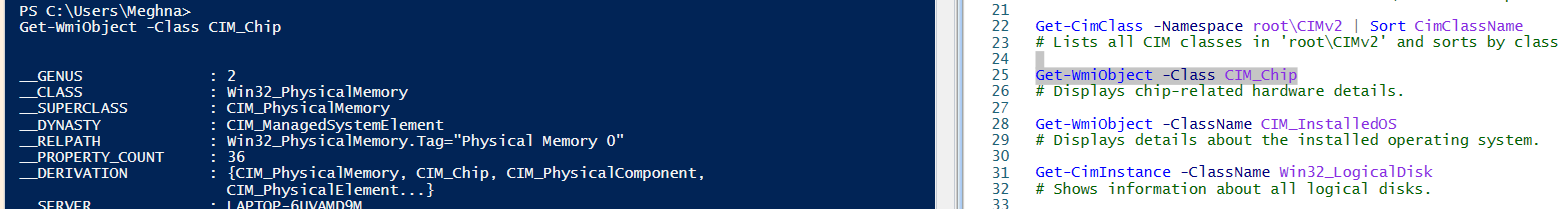
1. Get-CimClass -Namespace root\CIMv2 | Sort CimClassName

# Lists all CIM classes in 'root\CIMv2' and sorts by class name.



1. Get-WmiObject -Class CIM\_Chip

# Displays chip-related hardware details.



1. Get-WmiObject -ClassName CIM\_InstalledOS

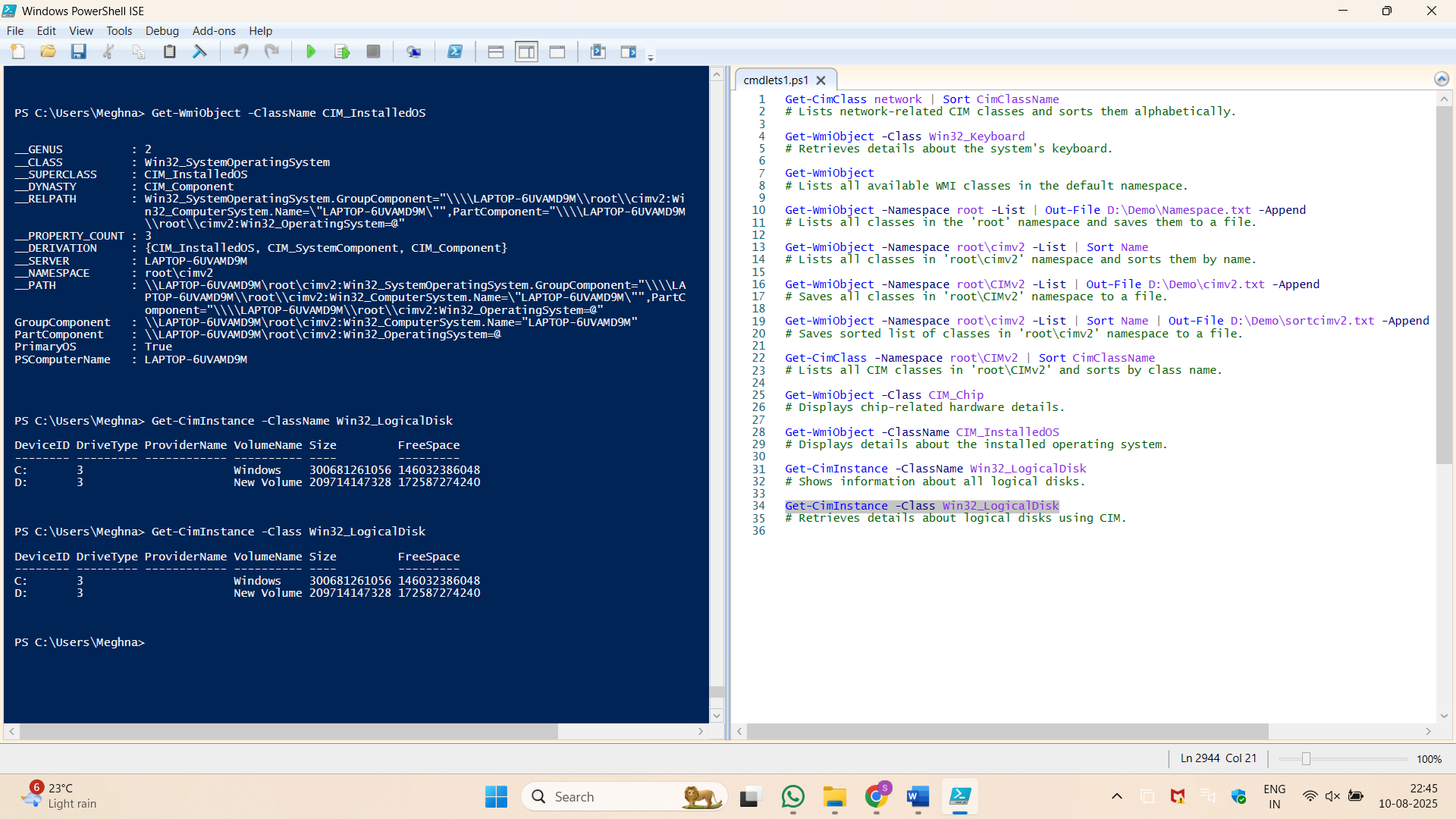
# Displays details about the installed operating system.

1. Get-CimInstance -ClassName Win32\_LogicalDisk

# Shows information about all logical disks.

1. Get-CimInstance -Class Win32\_LogicalDisk

# Retrieves details about logical disks using CIM.

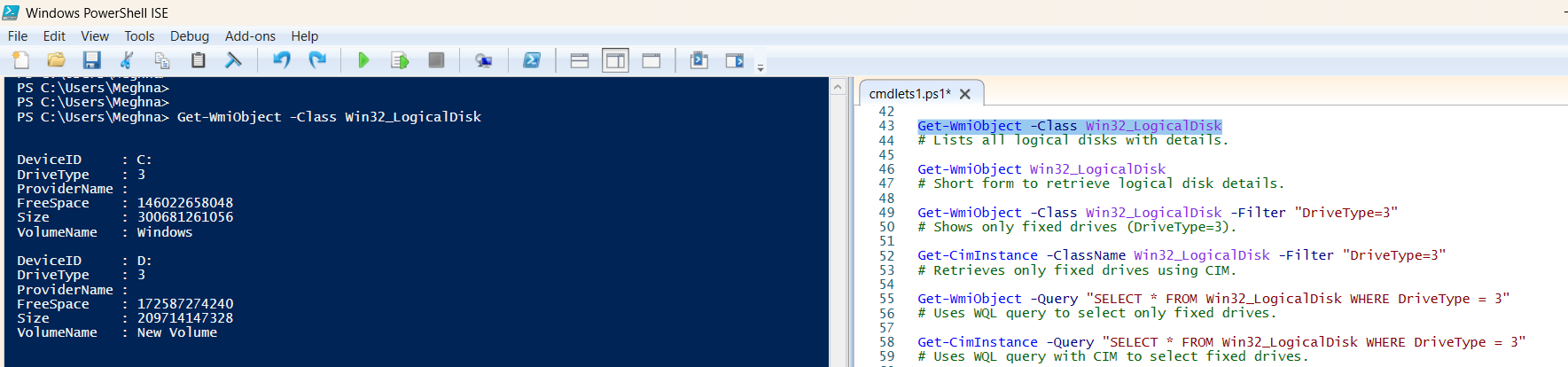


**2. Logical Disk Information**

**Purpose: Get details about logical disks and filter by type.**

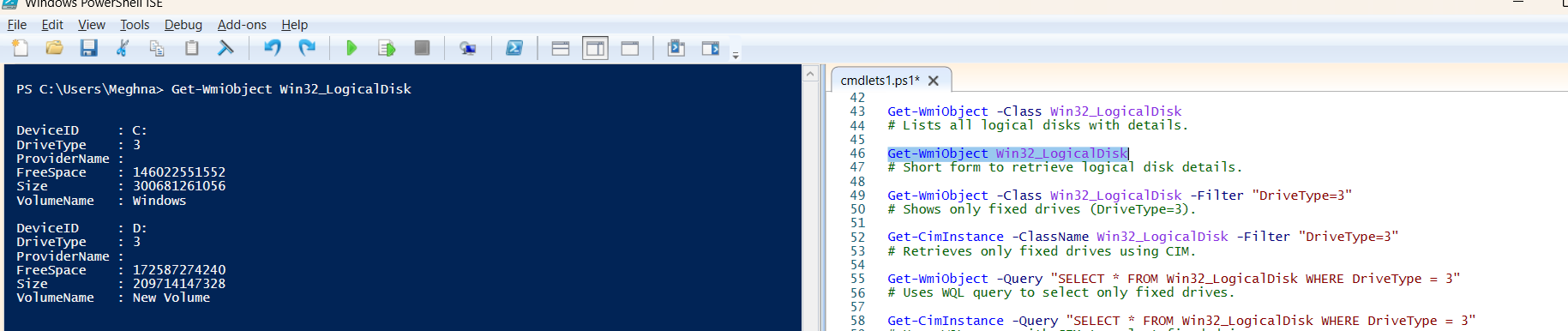
1. Get-WmiObject -Class Win32\_LogicalDisk

# Lists all logical disks with details.



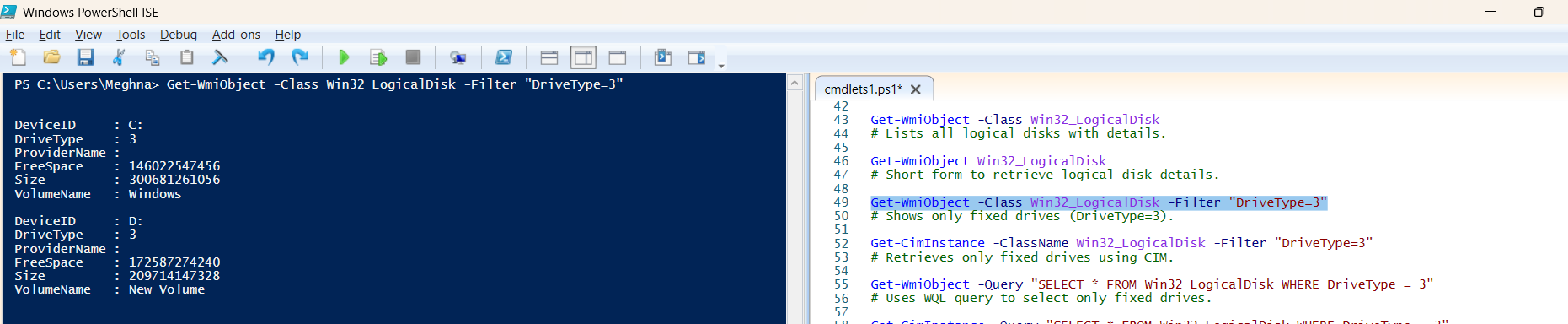
1. Get-WmiObject Win32\_LogicalDisk

# Short form to retrieve logical disk details.



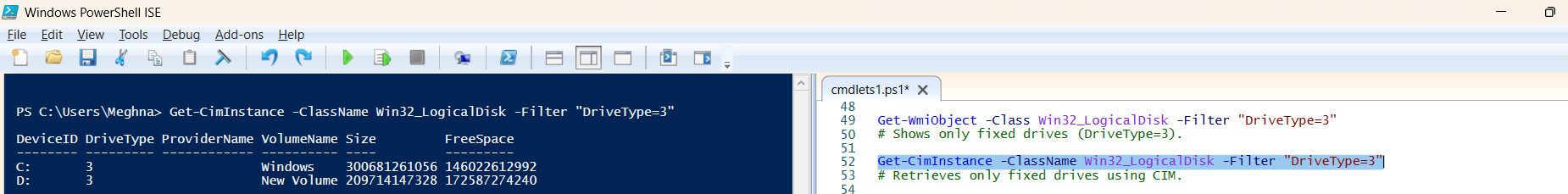
1. Get-WmiObject -Class Win32\_LogicalDisk -Filter "DriveType=3"

# Shows only fixed drives (DriveType=3).



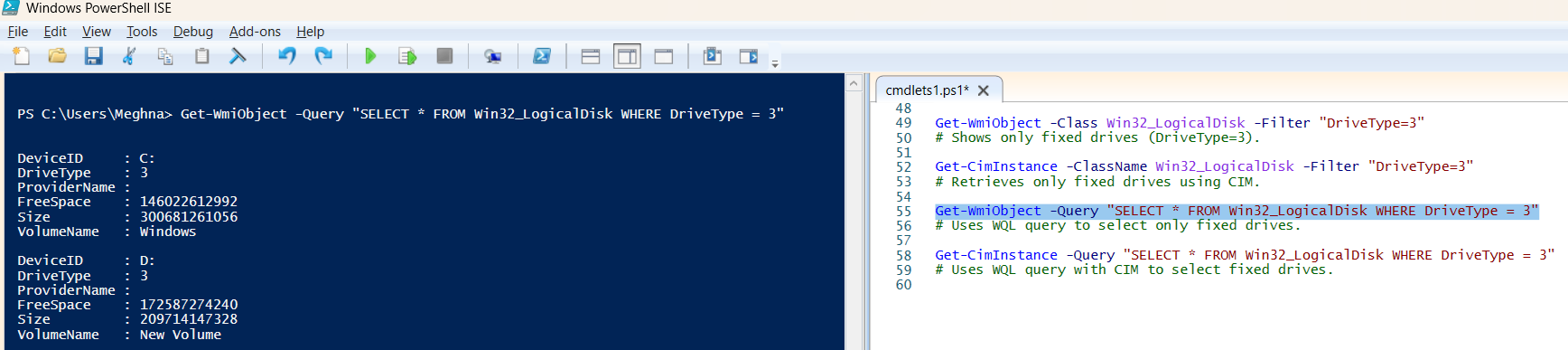
1. Get-CimInstance -ClassName Win32\_LogicalDisk -Filter "DriveType=3"

# Retrieves only fixed drives using CIM.



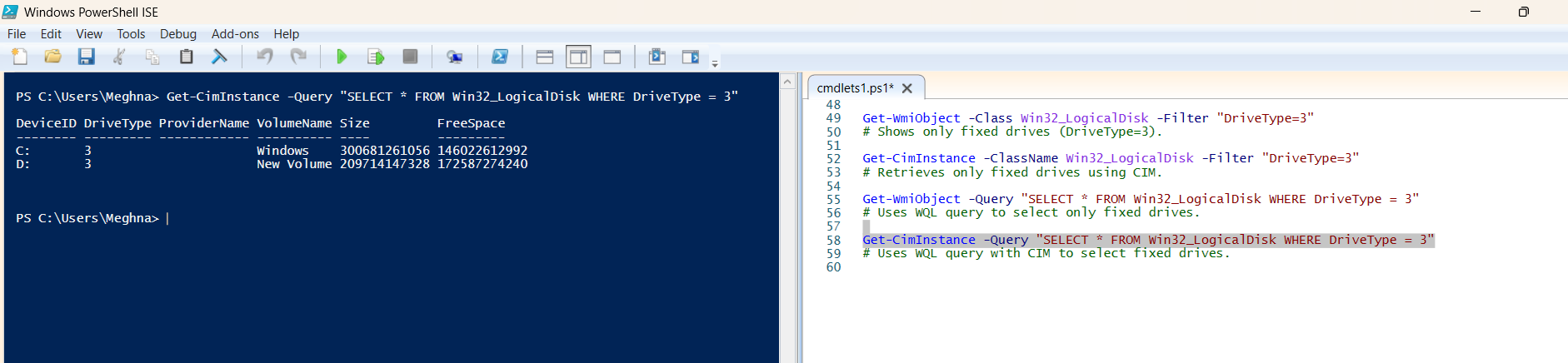
1. Get-WmiObject -Query "SELECT \* FROM Win32\_LogicalDisk WHERE DriveType = 3"

# Uses WQL query to select only fixed drives.



1. Get-CimInstance -Query "SELECT \* FROM Win32\_LogicalDisk WHERE DriveType = 3"

# Uses WQL query with CIM to select fixed drives.

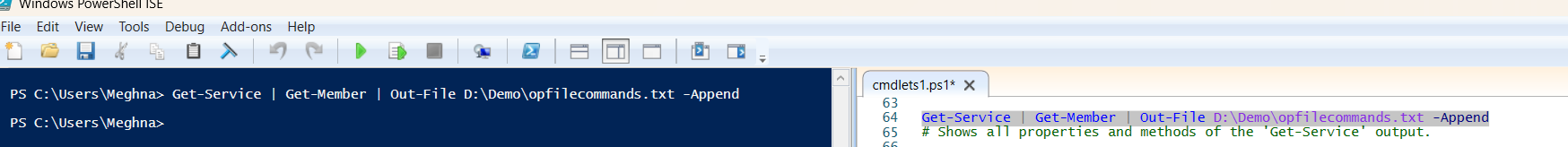


**3. Object Members & Command Info**

Purpose: Explore available properties and methods.

1. Get-Service | Get-Member | Out-File D:\Demo\opfilecommands.txt -Append

# Shows all properties and methods of the 'Get-Service' output.

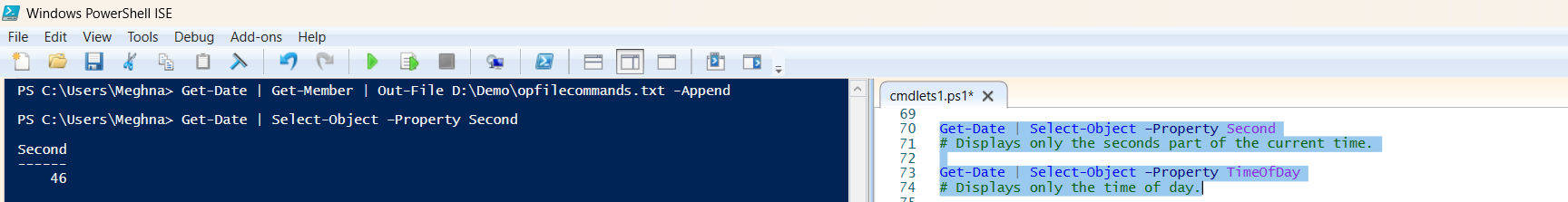


1. Get-Date | Get-Member | Out-File D:\Demo\opfilecommands.txt -Append

# Lists members (methods/properties) of the 'Get-Date' output.

1. Get-Date | Select-Object –Property Second

# Displays only the seconds part of the current time.



1. Get-Date | Select-Object –Property TimeOfDay

# Displays only the time of day.

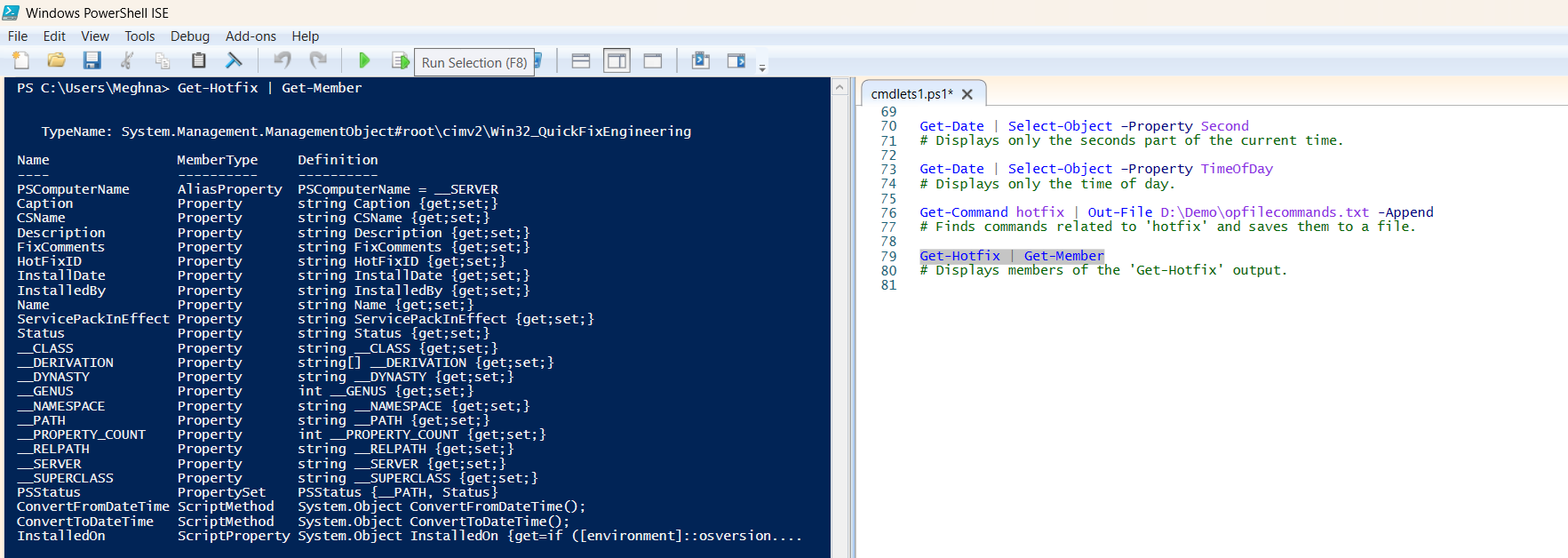


1. Get-Command hotfix | Out-File D:\Demo\opfilecommands.txt -Append

# Finds commands related to 'hotfix' and saves them to a file.

1. Get-Hotfix | Get-Member

# Displays members of the 'Get-Hotfix' output.

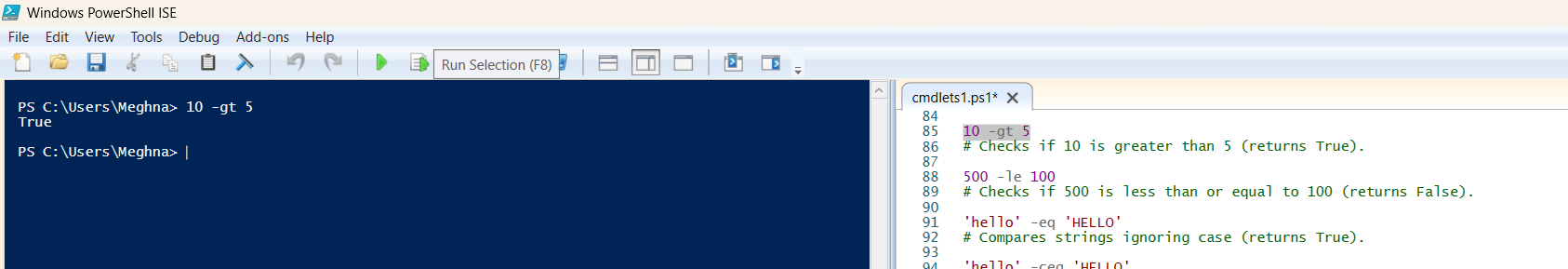


**4. Comparison Operators**

**Purpose: Compare values (case-sensitive and case-insensitive).**

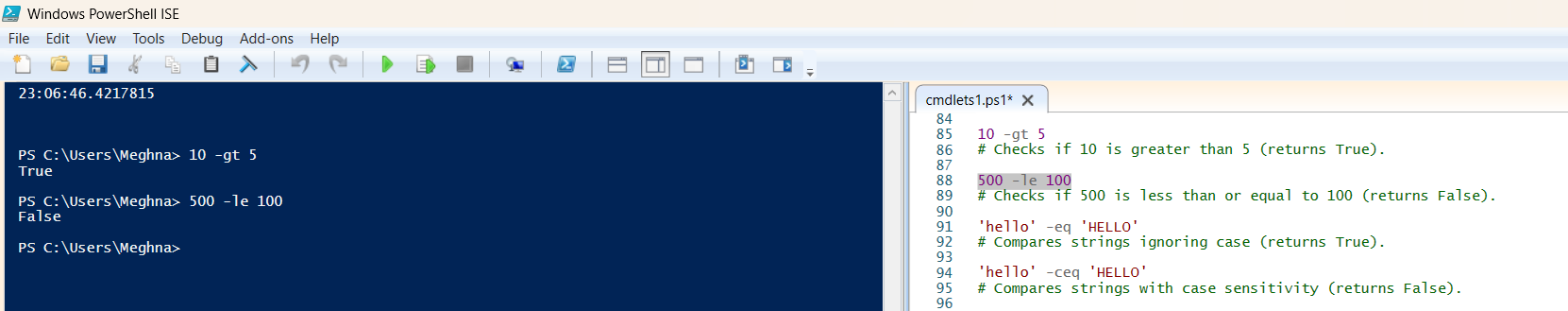
1. 10 -gt 5

# Checks if 10 is greater than 5 (returns True).



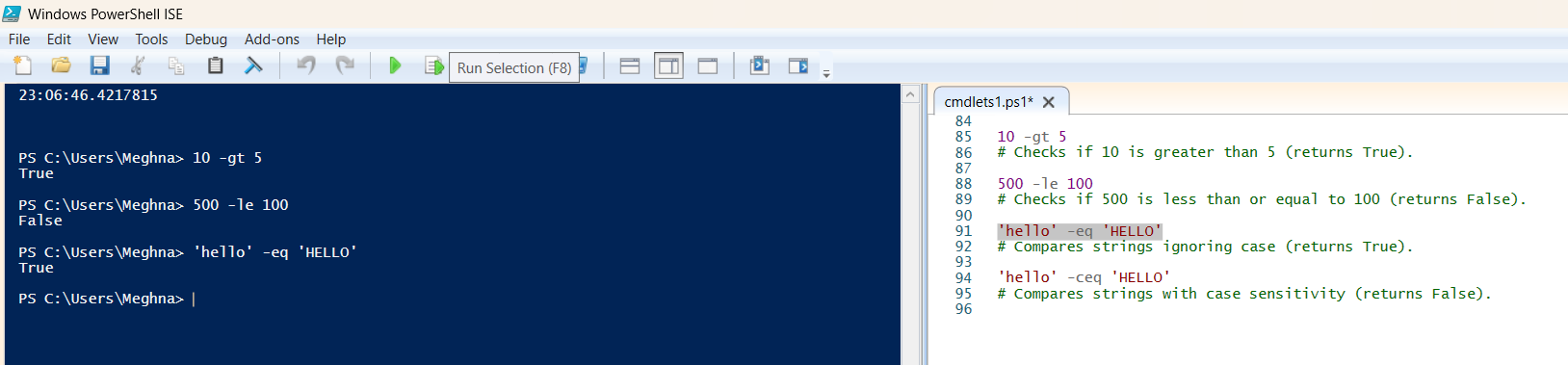
1. 500 -le 100

# Checks if 500 is less than or equal to 100 (returns False).



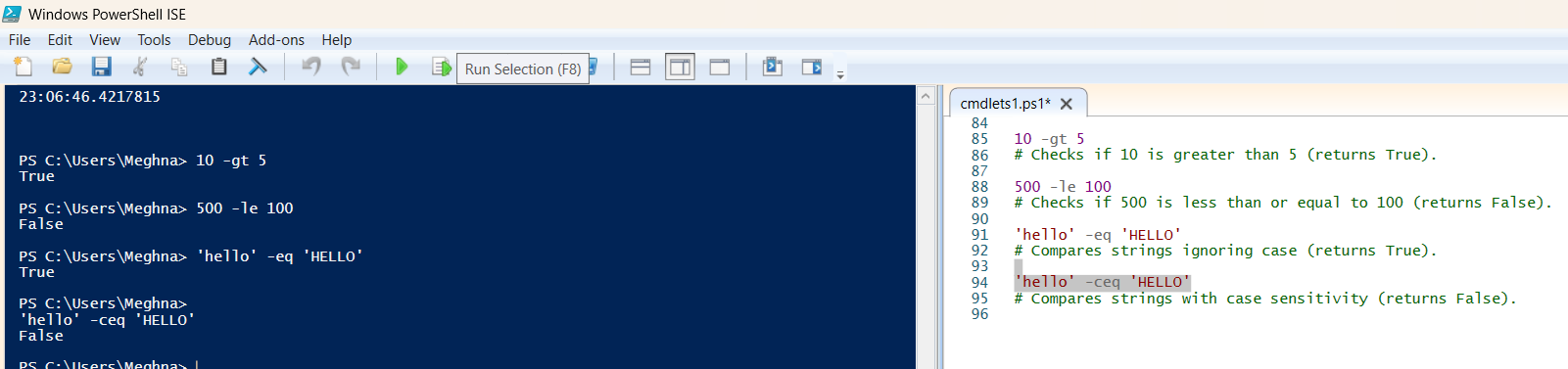
1. 'hello' -eq 'HELLO'

# Compares strings ignoring case (returns True).



1. 'hello' -ceq 'HELLO'

# Compares strings with case sensitivity (returns False).

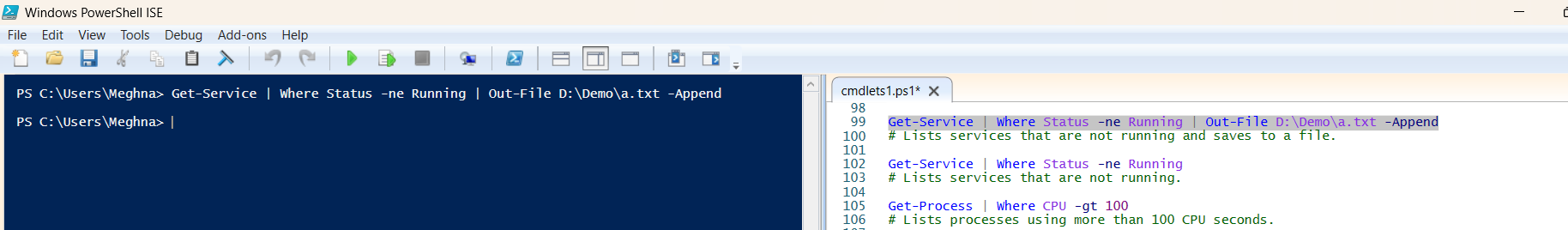


**5. Filtering Output**

**Purpose: Show only specific results based on conditions.**

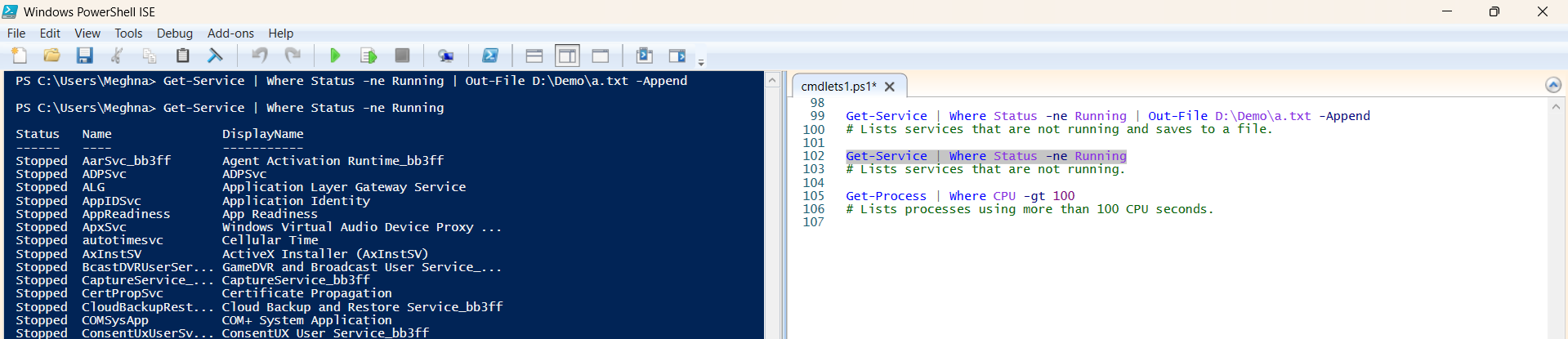
1. Get-Service | Where Status -ne Running | Out-File D:\Demo\a.txt -Append

# Lists services that are not running and saves to a file.



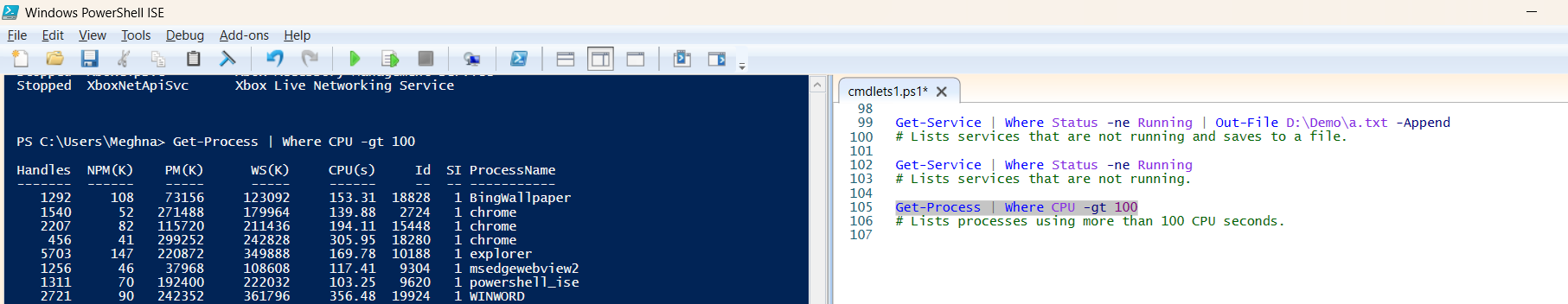
1. Get-Service | Where Status -ne Running

# Lists services that are not running.



1. Get-Process | Where CPU -gt 100

# Lists processes using more than 100 CPU seconds.

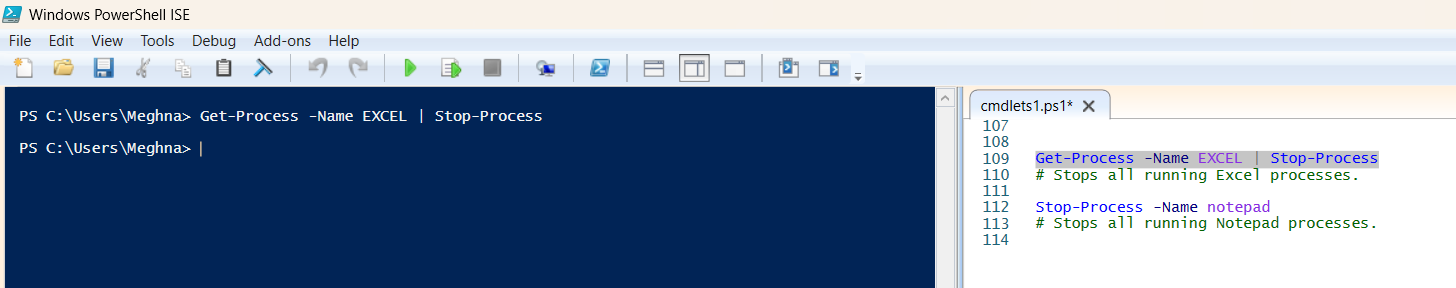


**6. Process Management**

**Purpose: Control running processes.**

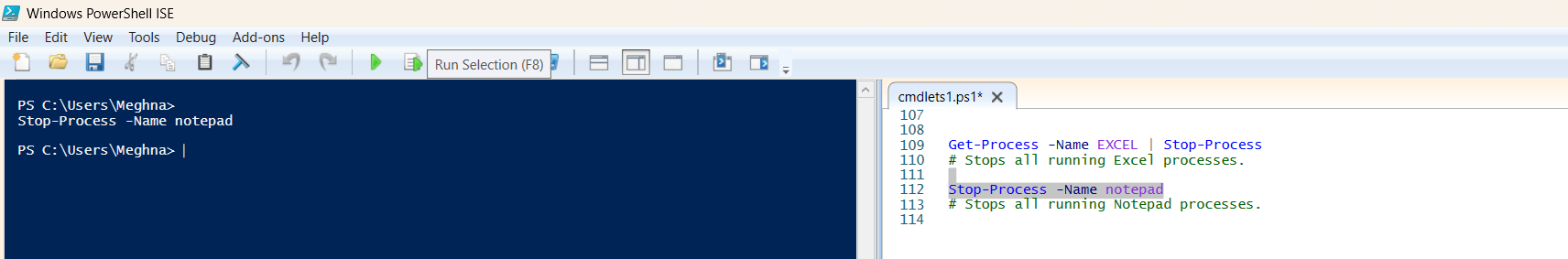
1. Get-Process -Name EXCEL | Stop-Process

# Stops all running Excel processes.



1. Stop-Process -Name notepad

# Stops all running Notepad processes.

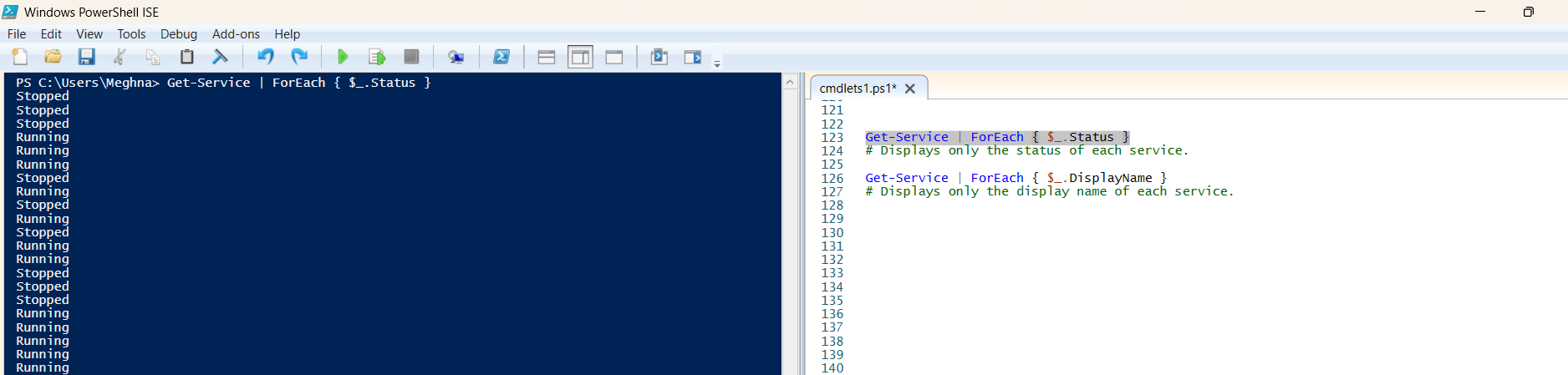


**7. ForEach Loop Examples**

**Purpose: Iterate through object properties.**

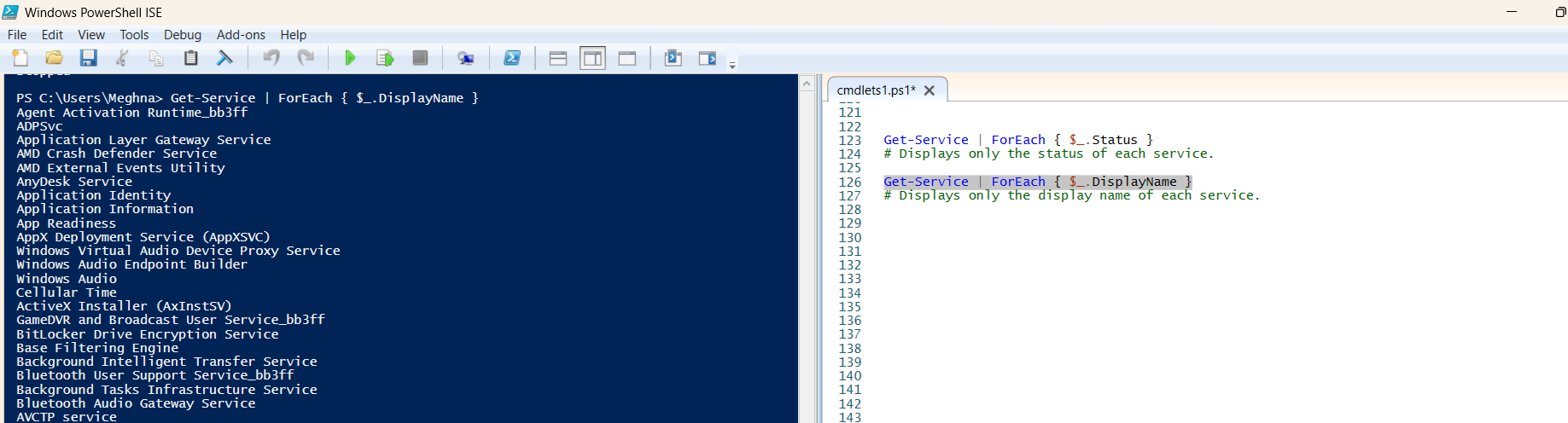
1. Get-Service | ForEach { $\_.Status }

# Displays only the status of each service.



1. Get-Service | ForEach { $\_.DisplayName }

# Displays only the display name of each service.

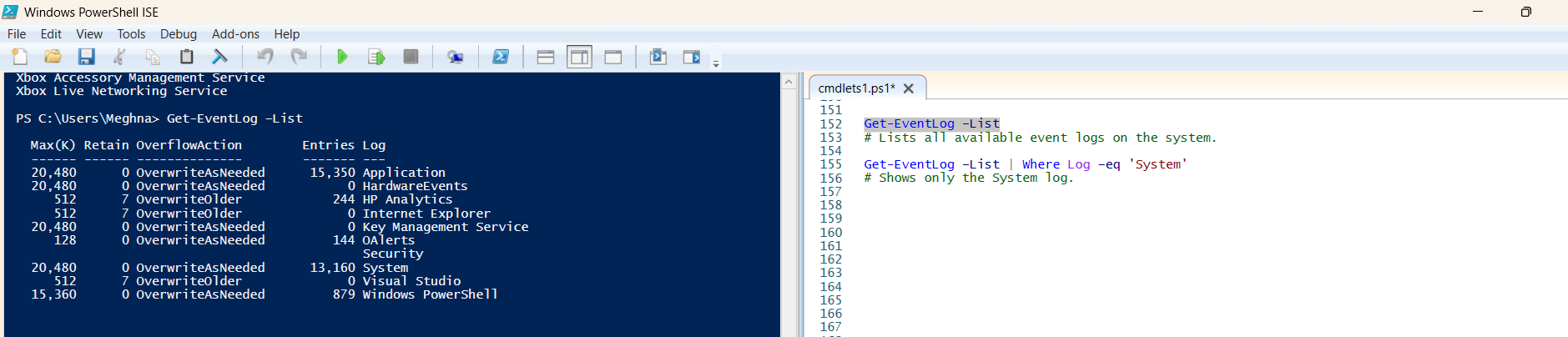


**8. Event Log Management**

**Purpose: Retrieve event logs.**

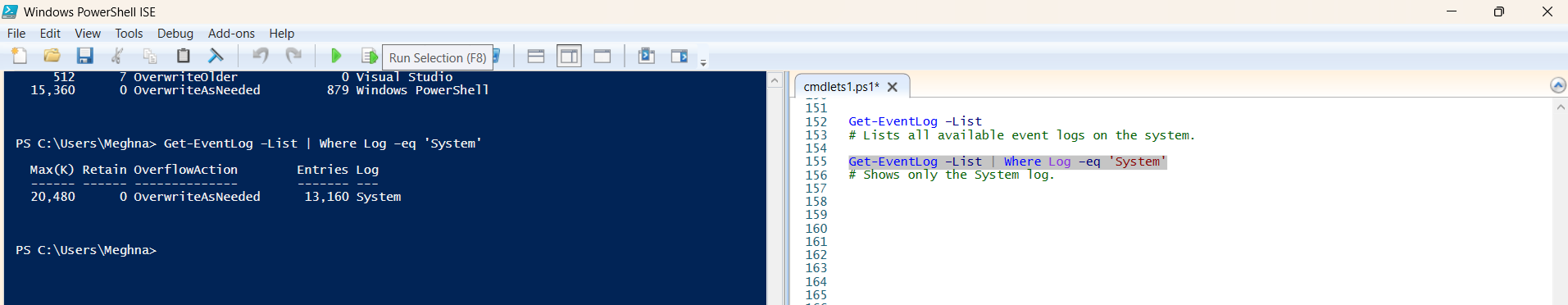
1. Get-EventLog –List

# Lists all available event logs on the system.



1. Get-EventLog –List | Where Log –eq 'System'

# Shows only the System log.

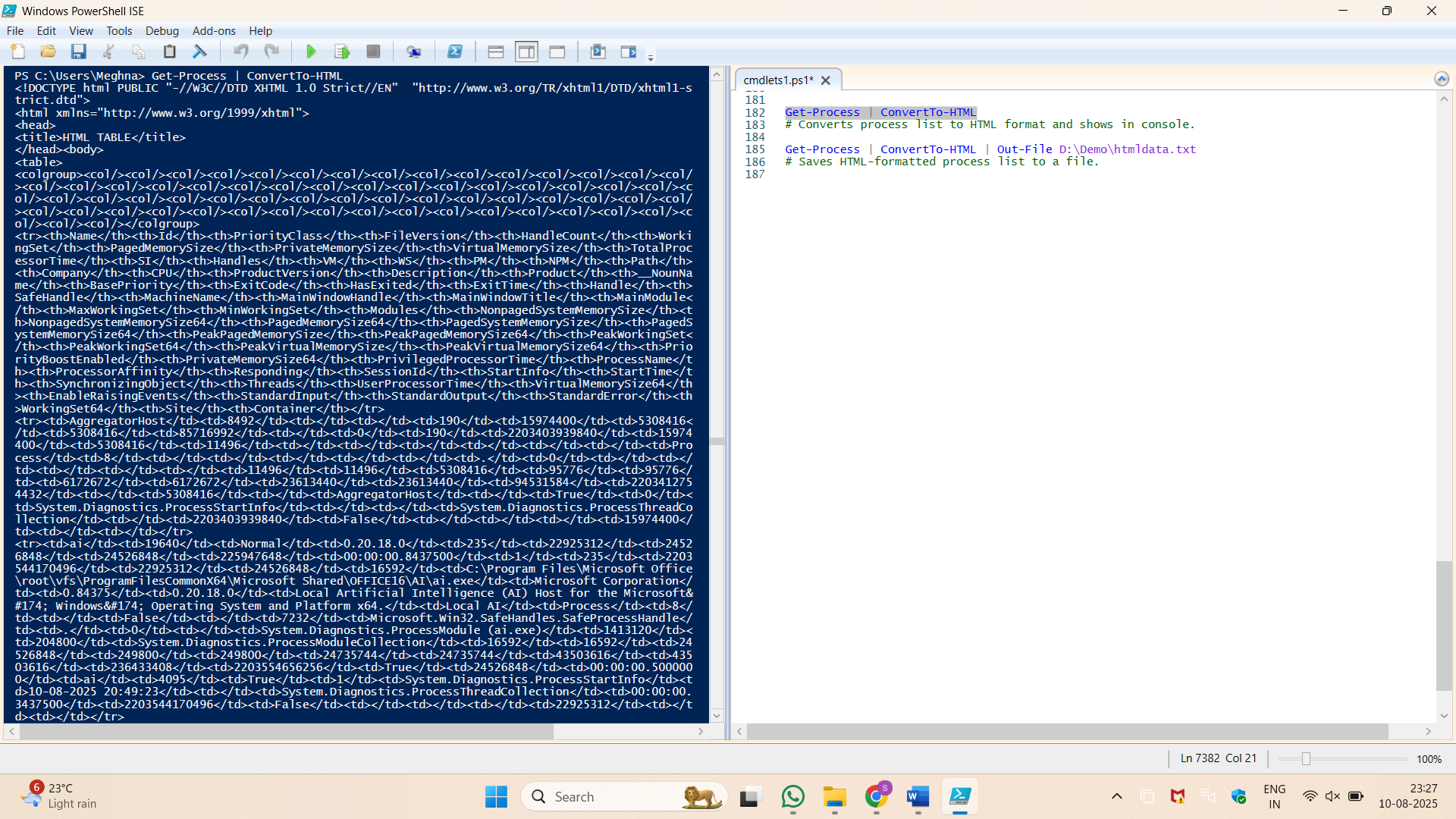


**9. Output Formatting**

**Purpose: Convert output to HTML format.**

1. Get-Process | ConvertTo-HTML

# Converts process list to HTML format and shows in console.



1. Get-Process | ConvertTo-HTML | Out-File D:\Demo\htmldata.txt

# Saves HTML-formatted process list to a file.

