**NAME – KARAN DAS**

**USER ID – 34763**

**MAIL-** [**karannnx20@gmail.com**](mailto:karannnx20@gmail.com)

**Date – 9th Aug 2025**

**Topic Assignment:**

**PowerShell Cmdlets Commands :-**

**Practice activity:**

**1. Get-NetIPConfiguration | Out-File E:\Powershell\karan**

Its Shows your computer’s network info, like IP address, gateway, and DNS.

**2. Get-PsDrive | Out-File "E:\Powershell\karan" -Append**

Lists all storage locations your computer can access (like C:, D:, registry, etc.).

**3. Get-NetIPConfiguration | Out-File "E:\Powershell\karan" -Append**

It shows your computer’s network details like:

* IP address
* Default gateway
* DNS server
* Network adapter name

**4. Get-Service| Out-File "E:\Powershell\karan" -Append**

Shows all services on your computer and their status (running, stopped).

**5. Get-PSDrive -PSProvider FileSystem | Out-File "E:\Powershell\karan" -Append**

Lists only drives from your file system (e.g., C:, D:), ignoring other providers like registry

**6. Get-Process**

Shows all the programs and background tasks running on your computer right now.

**7. Get-Help**

* Shows information about how to use a cmdlet.
* You can use it alone to see help topics or with a specific cmdlet name to see details.

**8. Get-Command**

* + Lists all available cmdlets, functions, and scripts on your system.
  + You can filter results to find specific commands.
  + Example:

Get-Command -Name \*service\*

Shows all commands related to "service".

**9 .Get-Process**

* + Displays all running programs and background processes.
  + Shows process name, ID, CPU usage, memory usage, etc.
  + Example:

Get-Process | Sort-Object CPU -Descending

Shows processes sorted by CPU usage (highest first).

**10. Get-Service**

* + Lists all Windows services with their status (Running, Stopped, etc.).
  + You can start/stop services using other cmdlets with the service name.
  + Example:

-Name "Spooler"

Shows the Print Spooler service status.

**11. Clear-Host**

* + Clears the PowerShell screen.
  + Shortcut: just type cls.

**12. Get-ChildItem**

* + Lists files and folders in the current or given location.
  + Similar to dir in Command Prompt, but with more features.
  + Example:

Get-ChildItem -Recurse

Lists all files including in subfolders.

**13. New-Item**

* + Creates a new file or folder.
  + Example:

New-Item -Path "C:\Test\sample.txt" -ItemType File

Creates a new text file.

**14. Copy-Item**

* + Copies files or folders from one location to another.
  + Example:

Copy-Item "C:\file.txt" -Destination "D:\Backup\"

**15. Move-Item**

* + Moves files or folders to a new location (also works for renaming).
  + Example:

Move-Item "C:\oldname.txt" "C:\newname.txt"

**16. Remove-Item**

* + Deletes files or folders.
  + Example:

Remove-Item "C:\Test\sample.txt"

**17. Test-Connection**

* + Sends ping requests to a computer or website to check connectivity.
  + Example:

Test-Connection google.com

**18. Get-NetAdapter**

* + Displays network adapters and their statuses (Up, Down, Disabled).

**19. Get-PSDrive**

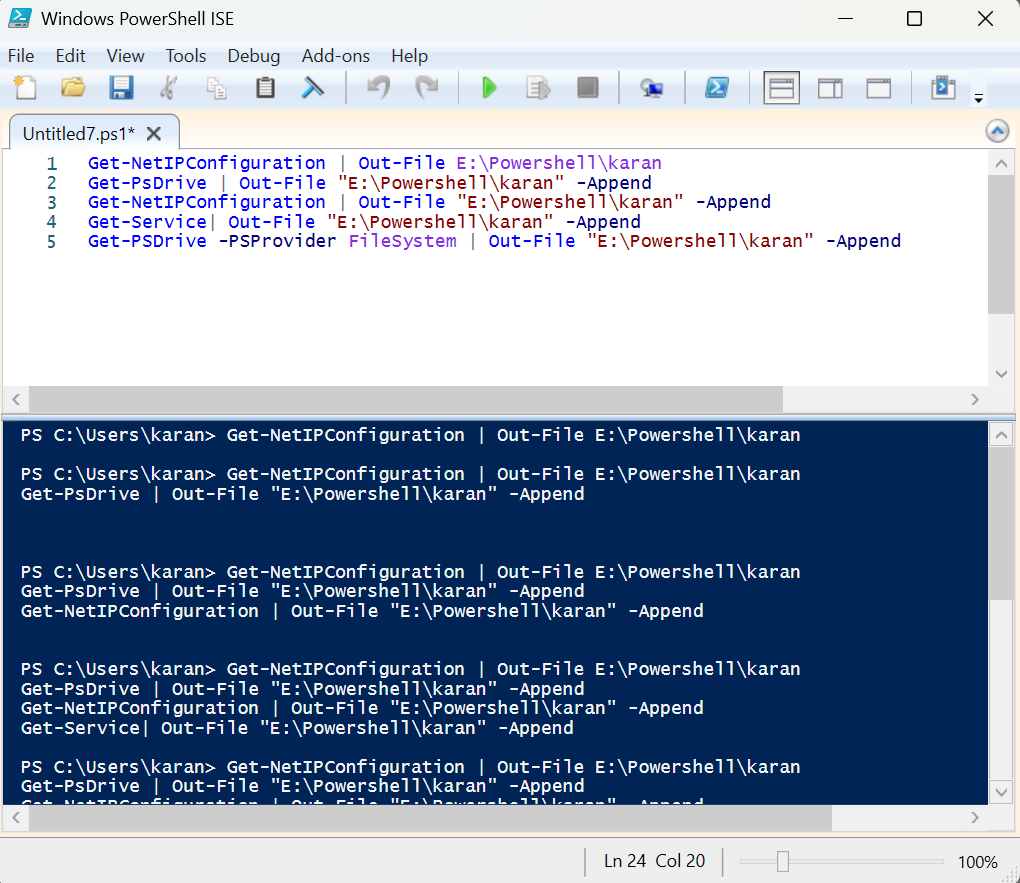
* + Lists all storage drives (C:, D:, network drives, registry hives).

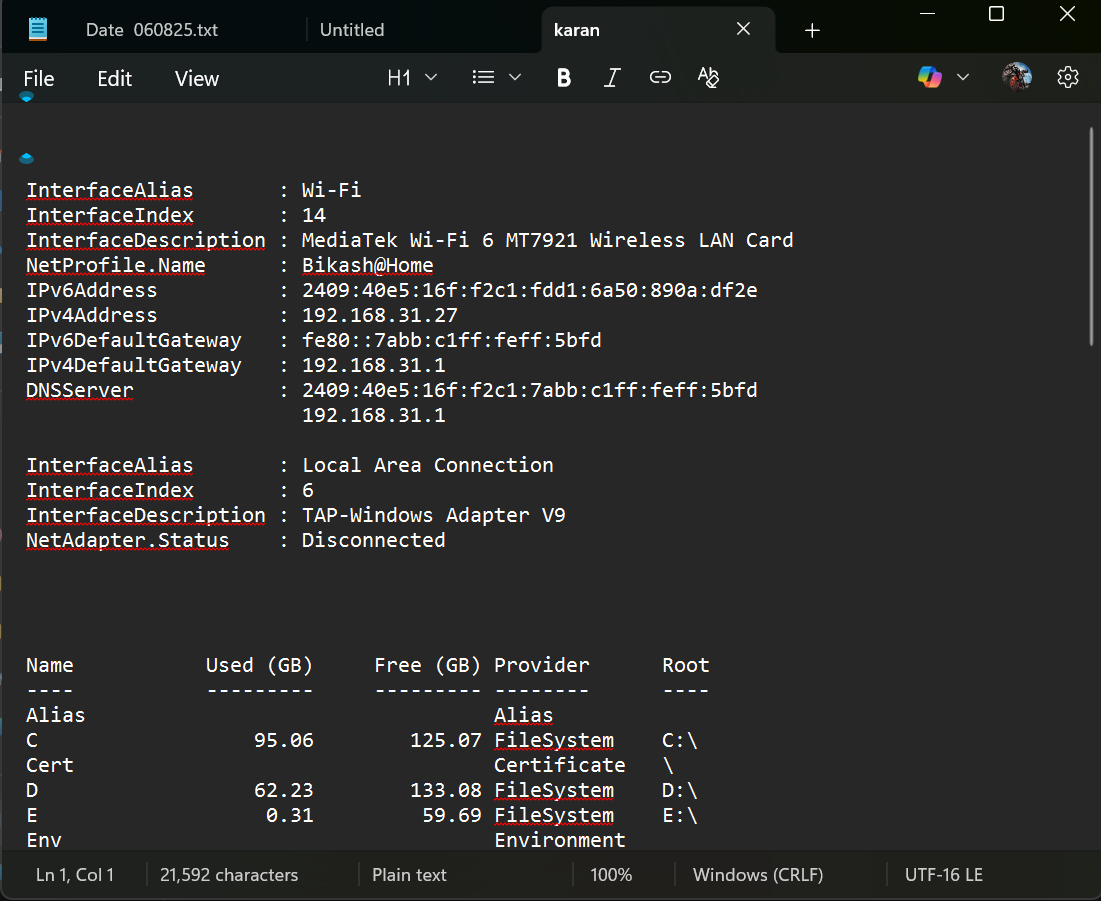
**20. Get-EventLog**

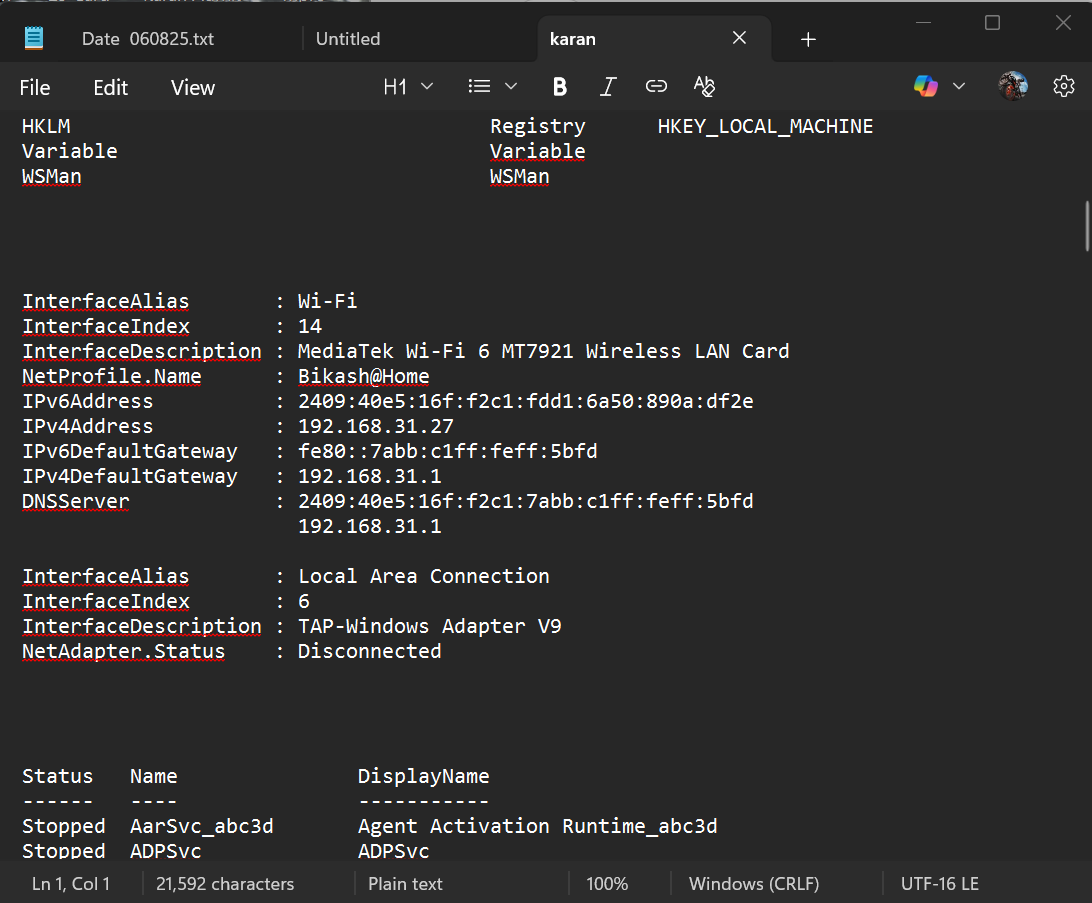
* + Reads Windows Event Logs for troubleshooting.

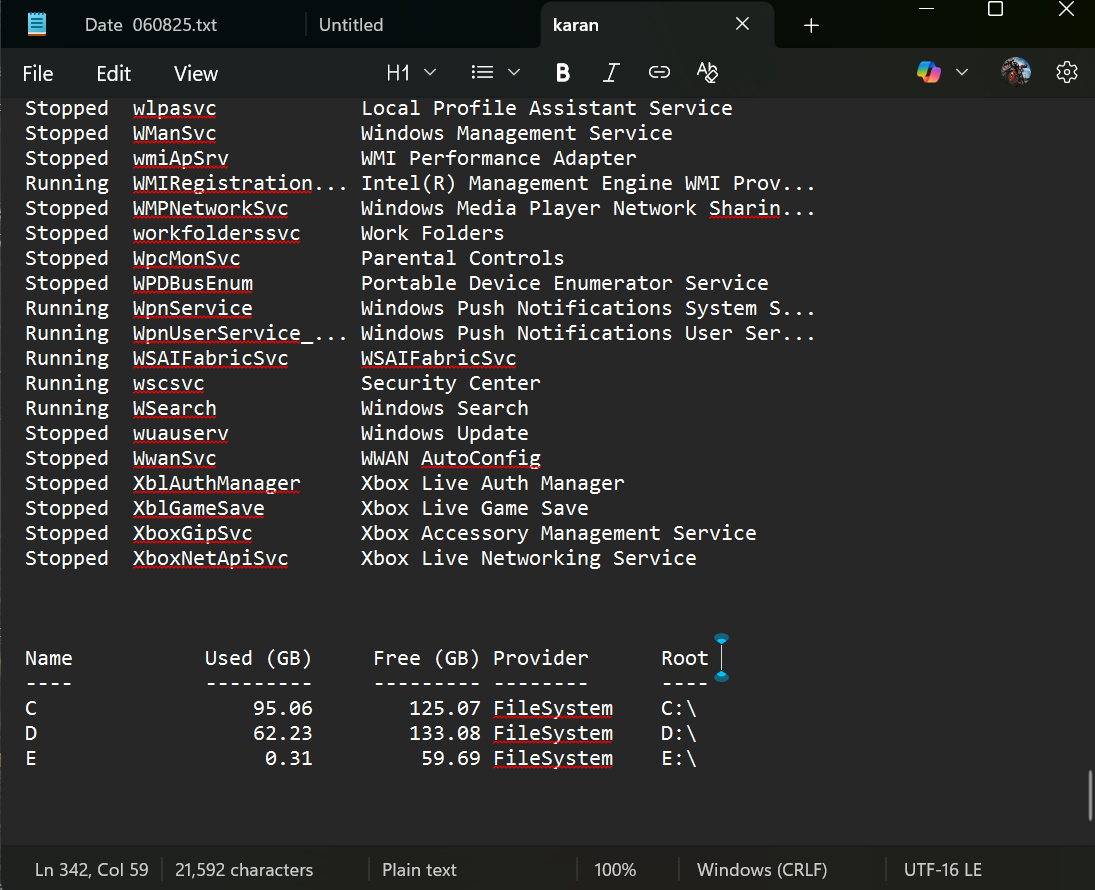
**21. Get-Date**

* + Shows the current date and time.
  + You can also format the output.



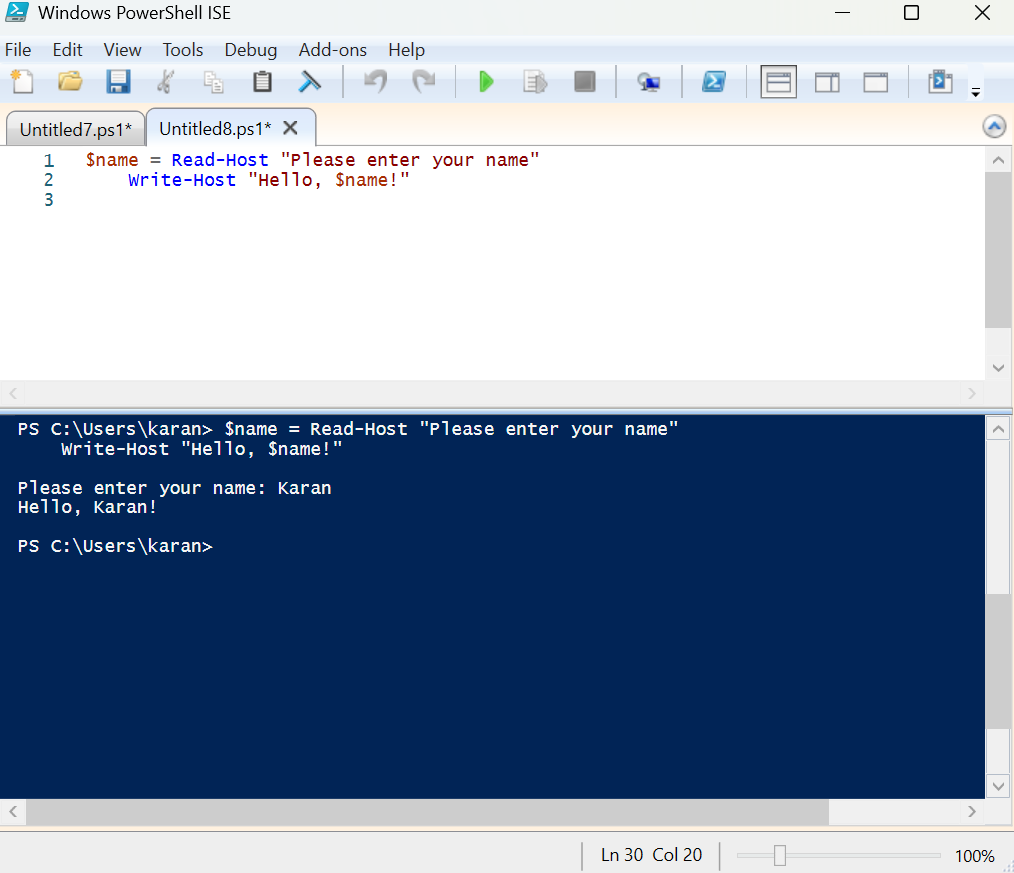






**6. $name = Read-Host "Please enter your name"**

**Write-Host "Hello, $name!"**

****

**7. "My name is $name and I am $age years old" -f $name, $age**

**8. Get-Process | Sort-Object CPU -Descending | Select-Object -First 5 | Select-Object Name, CPU**

**9. $fruits = "banana","apple"**

**$fruits[0]**

**$fruits+="kiwi"**

**$fruits[2]**

**10 . $age = 20**

**if ($age -ge 22) {**

**write-Host "Age is greater"**

**}**

**$age = 25**

**if ($age -ge 18) {**

**Write-Host "Adult"**

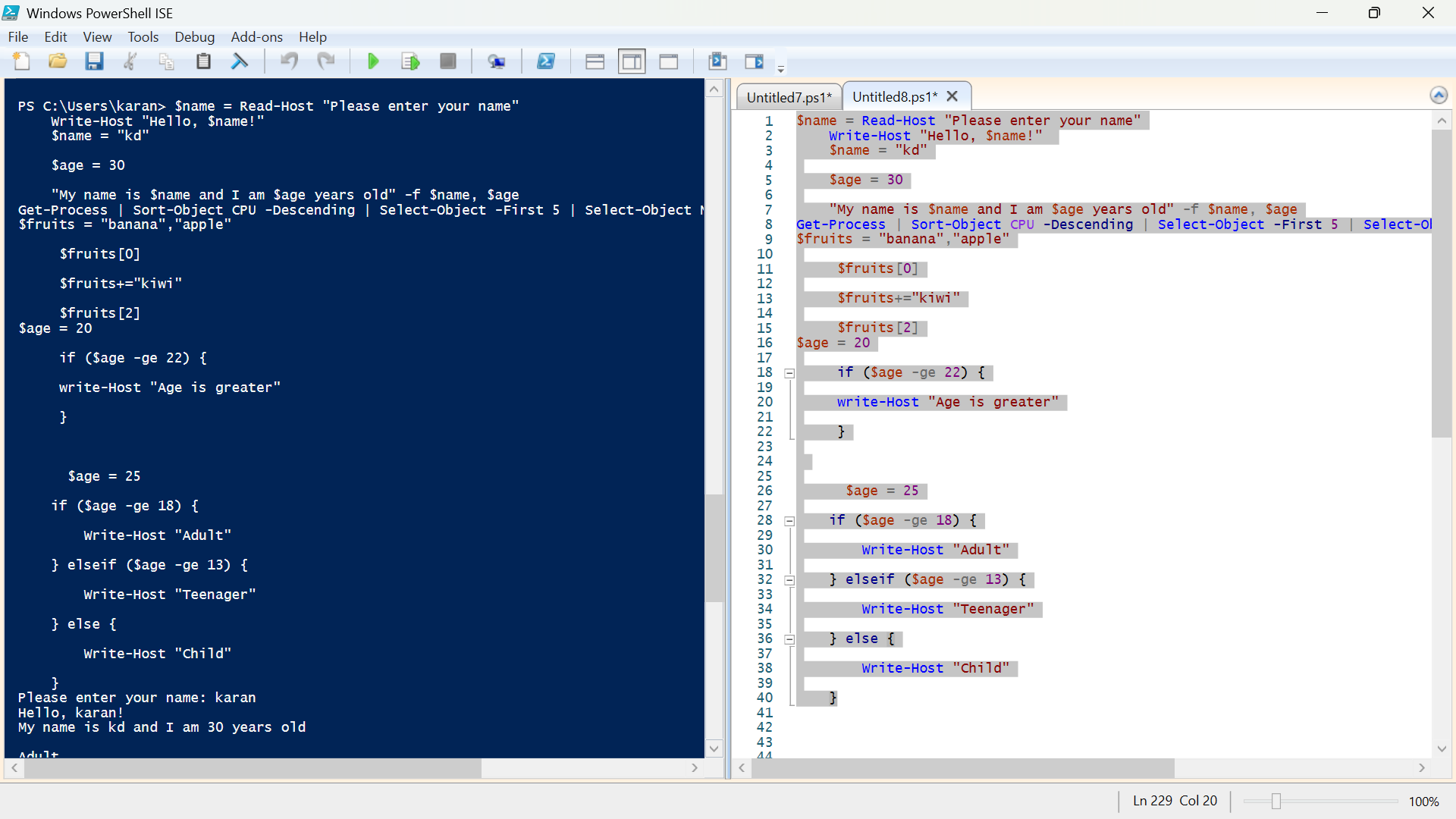
**} elseif ($age -ge 13) {**

**Write-Host "Teenager"**

**} else {**

**Write-Host "Child"**

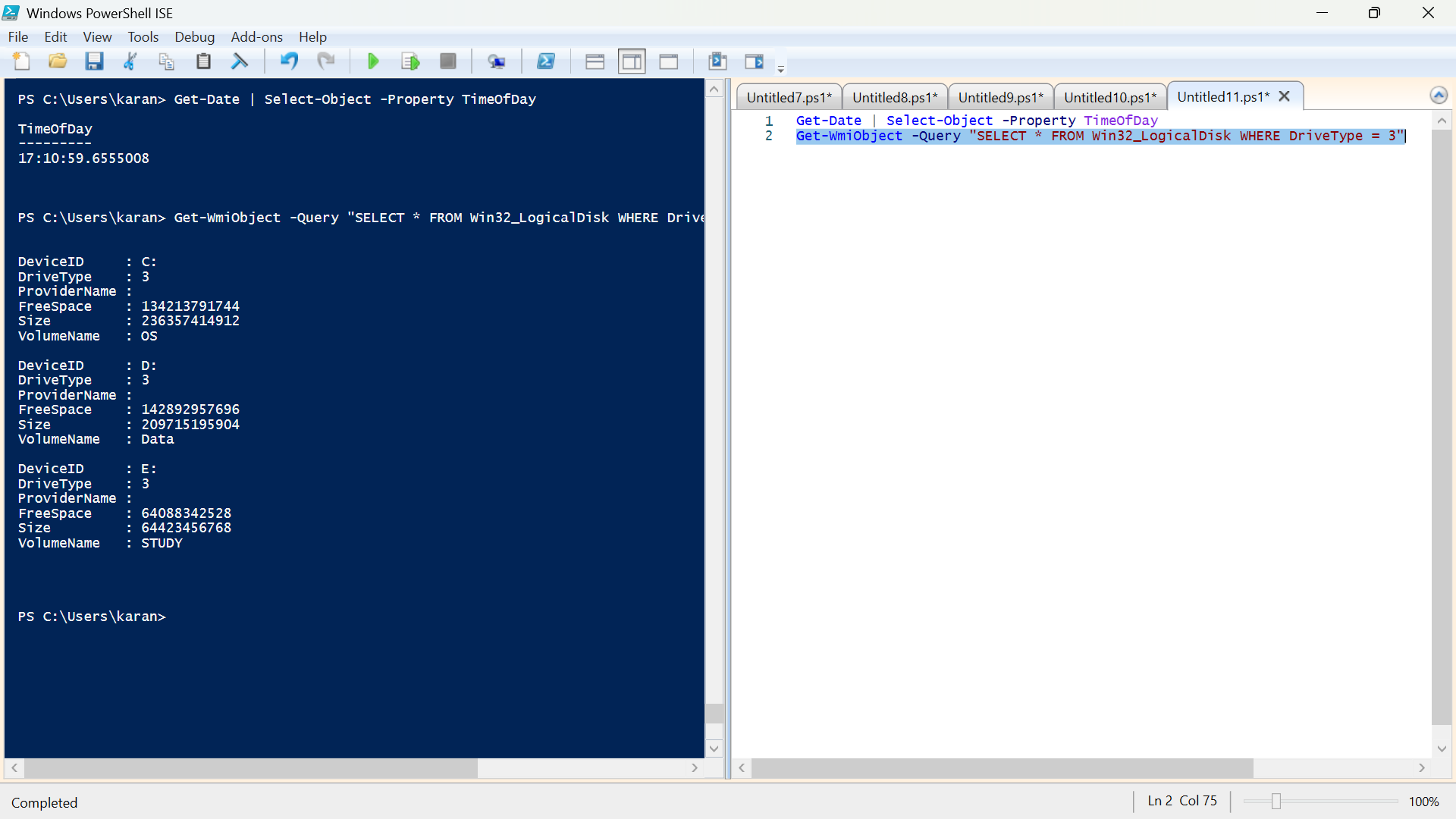
**}**



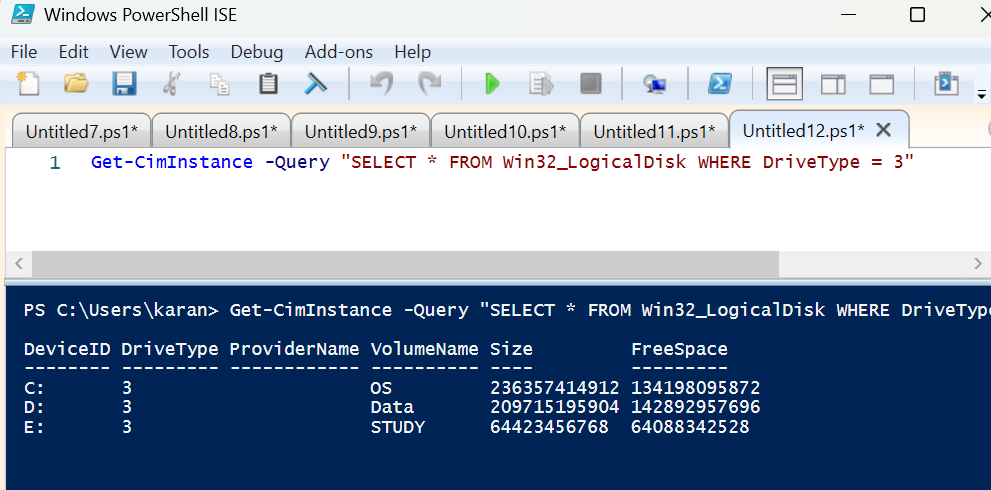
**Get-Date | Select-Object -Property TimeOfDay**

This command gets the system’s current date and time, then displays the TimeOfDay property as a TimeSpan value showing the elapsed time since midnight, including hours, minutes, seconds, and fractions, often used for time calculations or performance measurements.

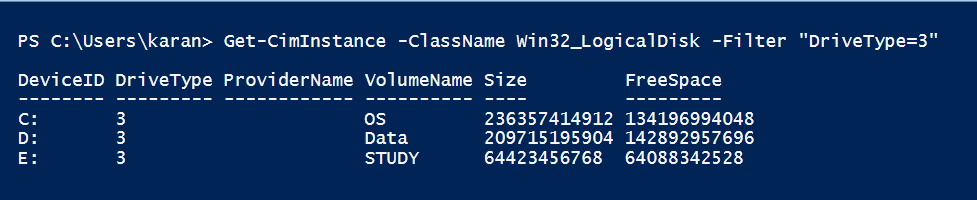
**Get-WmiObject -Query "SELECT \* FROM Win32\_LogicalDisk WHERE DriveType = 3"**



**Get-CimInstance -Query "SELECT \* FROM Win32\_LogicalDisk WHERE DriveType = 3"**  
Executes a CIM query returning all fixed local disks with their properties, similar to the WMI query but using the newer CIM cmdlet interface.

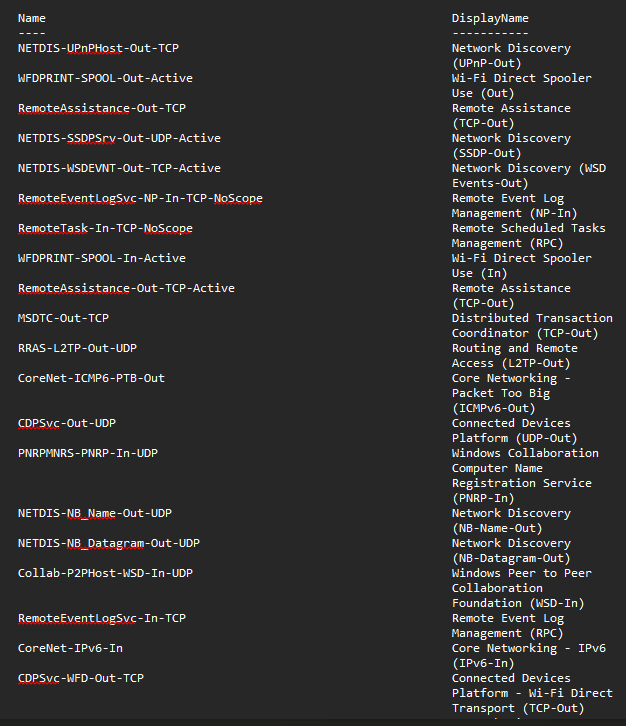


**Get-CimInstance -ClassName Win32\_LogicalDisk -Filter "DriveType=3**



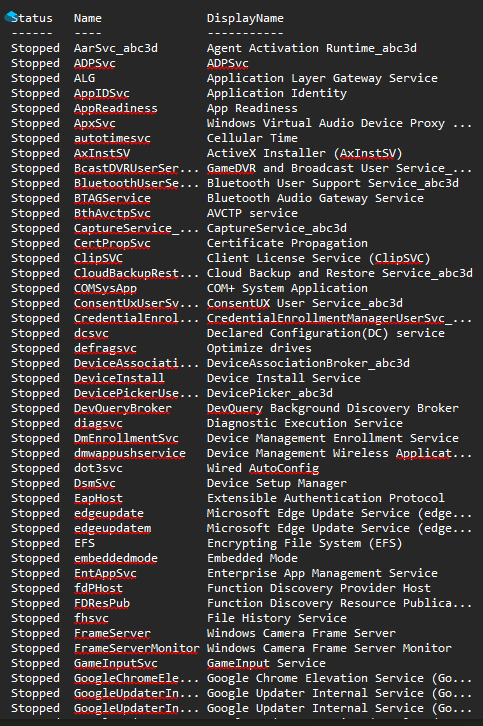
**Get-NetFirewallRule | Format-Table -wrap | Out-File "E:\Powershell\karandas.txt" -Append**

Lists all firewall rules (enabled and disabled), formats them in a word‑friendly table with wrapped text, and appends this output to abc.txt. Useful for keeping a complete, well‑formatted firewall configuration record in a shareable document



**Get-Service | Where status -ne Running | Out-File "E:\Powershell\karandass.txt" -Append**

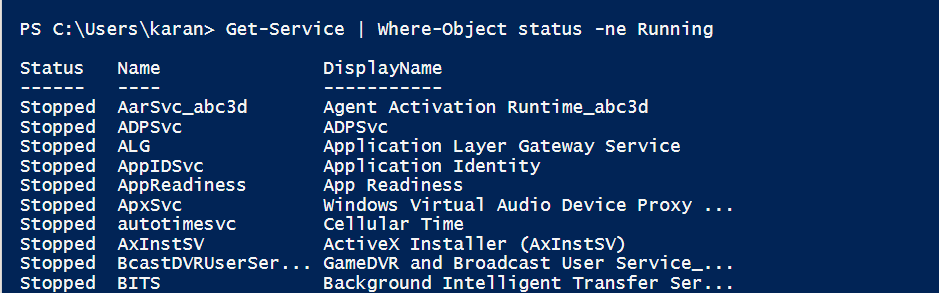
This command finds all services that are not running, then appends this list of stopped or paused services to the specified text file, useful for keeping a log of non-active services for monitoring or troubleshooting.



**Get-Service | Where-Object status -ne Running**

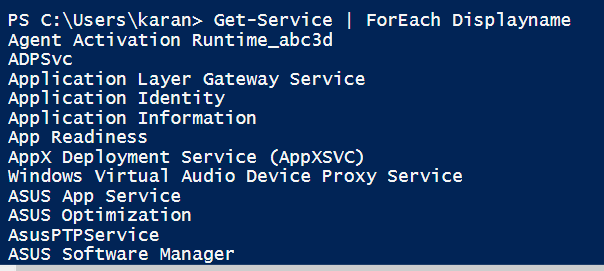
Get-Service :- Lists all services on your system.

Where-Object Status -ne 'Running' :- Filters the list to show only services not equal (-ne) to "Running".



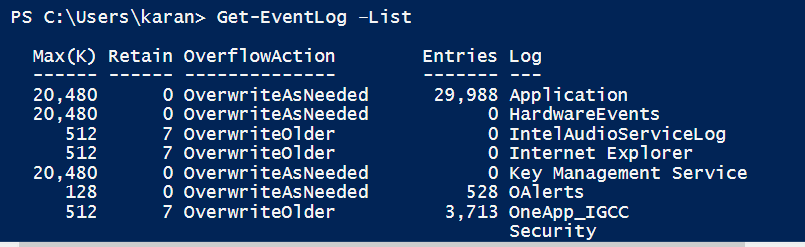
**Get-Service | ForEach Displayname**

This lists the friendly names of all services running or stopped on the system.



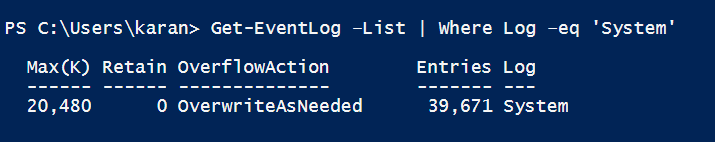
**Get-EventLog –List**

Shows a summary of all event logs present on the computer, including log names, sizes, and retention policies. This helps identify available logs for monitoring and troubleshooting Windows events.



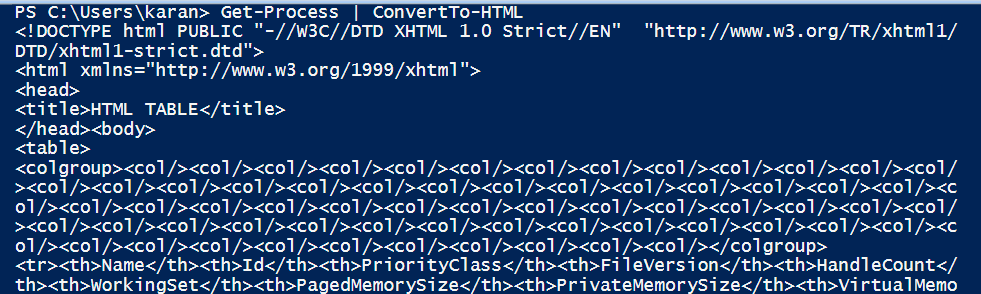
**Get-EventLog –List | Where Log –eq 'System'**

Filters the list of event logs to show only the "System" log, which records system-level events such as driver failures or hardware changes, useful for focused system troubleshooting.

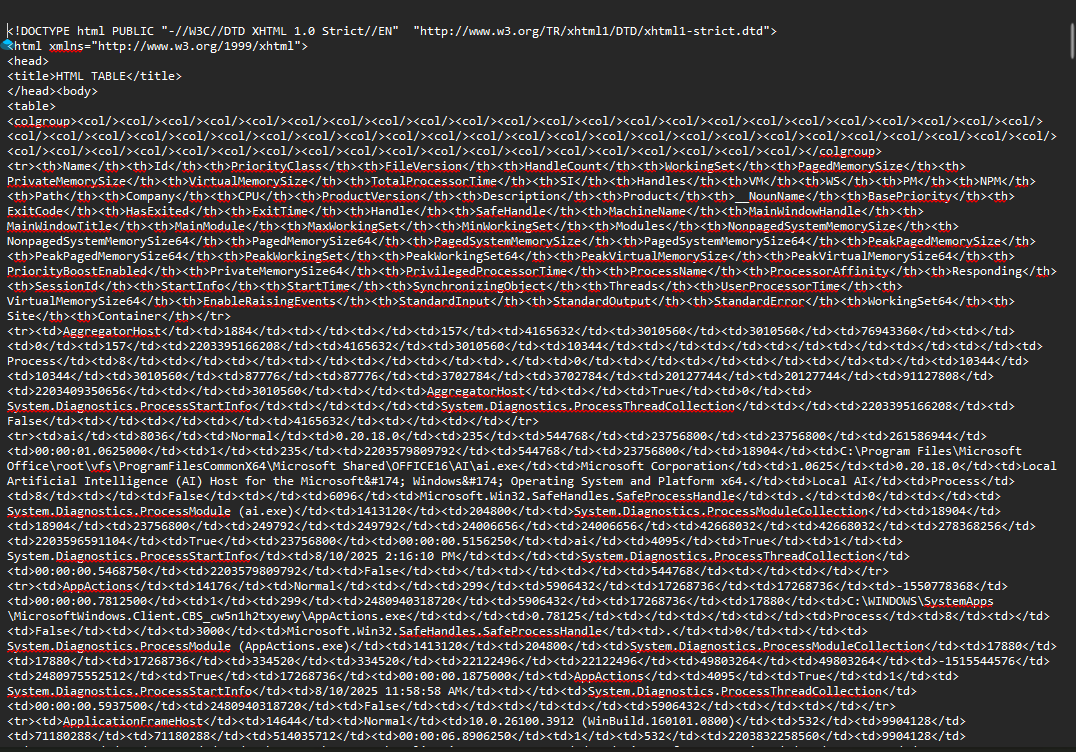


**Get-Process | ConvertTo-HTML**

It convert all process into html .

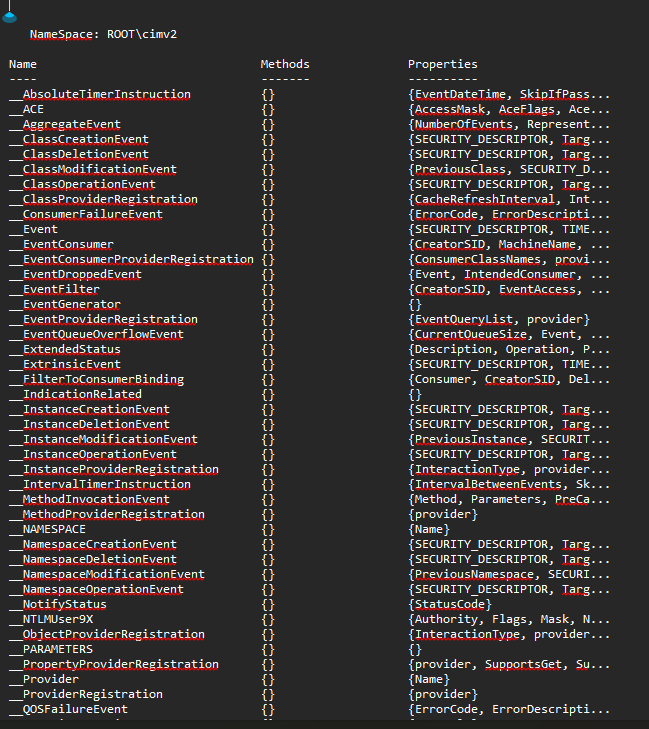


**Get-Process | ConvertTo-HTML | Out-File E:\Powershell\karandass1.txt"**

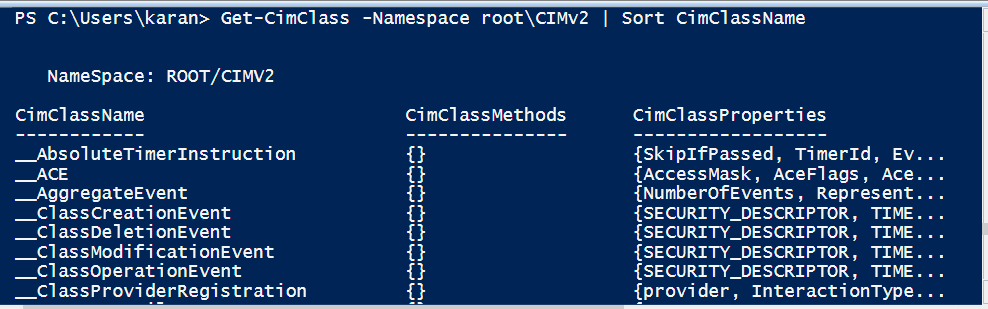
****

**Get-WmiObject -Namespace root\cimv2 -List | Sort Name | Out-File “E:\Powershell\karandass2.txt" -Append**

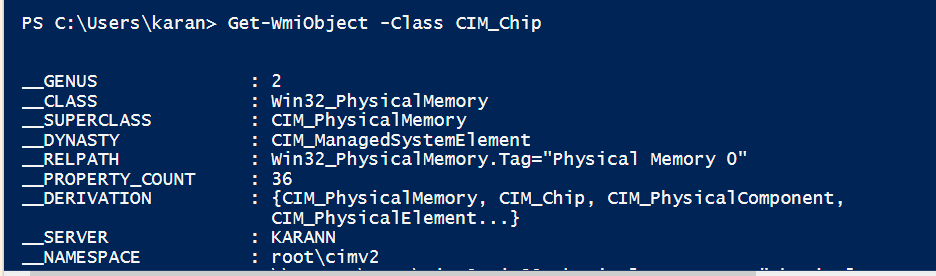
Fetches all WMI classes from root\cimv2, sorts them by name alphabetically, and appends them into the specified file. This makes it easy to maintain an alphabetically arranged reference list of CIMv2 classes for later review or administrative tasks.

****

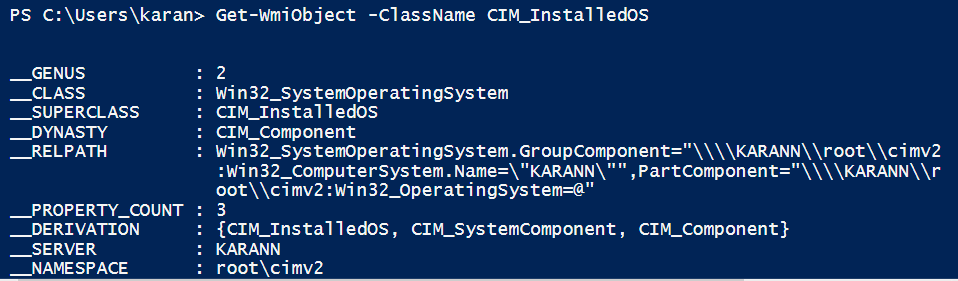
**Get-CimClass -Namespace root\CIMv2 | Sort CimClassName**  
This command retrieves all CIM classes available in the root\CIMv2 namespace and sorts them alphabetically by their class name. It helps explore system classes used for device and configuration management.



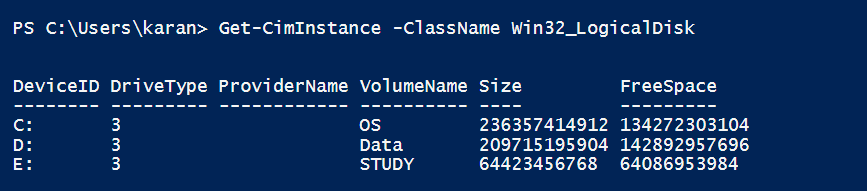
**Get-WmiObject -Class CIM\_Chip**  
This retrieves information about hardware chips on your system like processors and chipsets, using the WMI CIM\_Chip class. It provides details such as manufacturer, version, and device ID for chip-level hardware info.



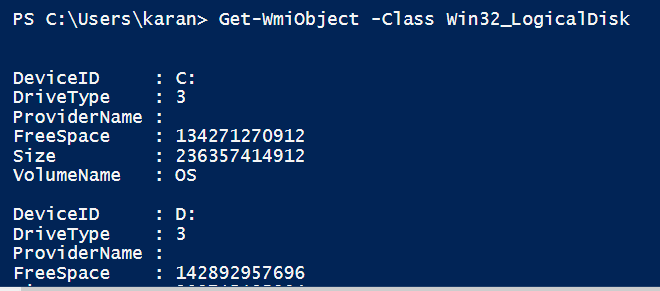
**Get-WmiObject -ClassName CIM\_InstalledOS**  
This gets information about the installed operating system on the computer by querying the WMI class CIM\_InstalledOS, including OS version, manufacturer, and installation details.



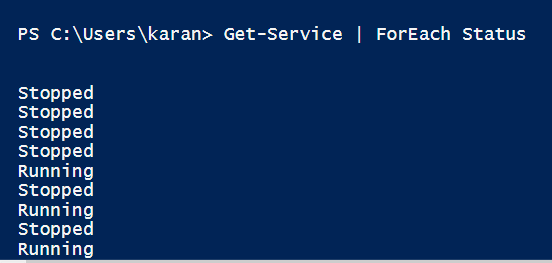
**Get-CimInstance -ClassName Win32\_LogicalDisk**  
Retrieves instances of all logical disks (like hard drives and partitions) on the system using CIM cmdlets. It provides data such as device ID, size, free space, and drive type.



**Get-WmiObject -Class Win32\_LogicalDisk**  
Performs a similar query using WMI to get all logical disk instances with properties like disk size, free space, and drive letter. Useful for disk and storage management.



**Get-Service | ForEach Status**

****