Lab 2 – PowerShell Basics & System Administration

Task 1: Exploring PowerShell Commands

▨ Step 1: Identify available PowerShell commands

Command:

```
Get-Command
```

Explanation:

Displays all available cmdlets, functions, workflows, aliases, and applications registered in the current PowerShell session.

Screenshot:

```
PS C:\Users\gkeymole> Get-Command

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           Add-AppPackage                                     2.0.1.0    Appx
Alias           Add-AppPackageVolume                               2.0.1.0    Appx
Alias           Add-AppProvisionedPackage                          3.0        Dism
Alias           Add-MsixPackage                                    2.0.1.0    Appx
Alias           Add-MsixPackageVolume                              2.0.1.0    Appx
Alias           Add-MsixVolume                                     2.0.1.0    Appx
Alias           Add-ProvisionedAppPackage                          3.0        Dism
Alias           Add-ProvisionedAppSharedPackageContainer           3.0        Dism
Alias           Add-ProvisionedAppxPackage                         3.0        Dism
```

▨ Step 2: Retrieve detailed help for a specific command

Command:

```
Get-Help Get-Process -Full
```

Explanation:

Shows the complete help documentation for the Get-Process cmdlet, including syntax, parameters, remarks, and examples.

Screenshot:

```
PS C:\Users\gkeymole> Get-Help Get-Process -Full

NAME
    Get-Process

SYNOPSIS
    Gets the processes that are running on the local computer or a remote computer.


SYNTAX
    Get-Process [[-Name] <System.String[]>] [-ComputerName <System.String[]>] [-Fil
    [<CommonParameters>]
```

▨ Step 3: List all properties and methods of an object

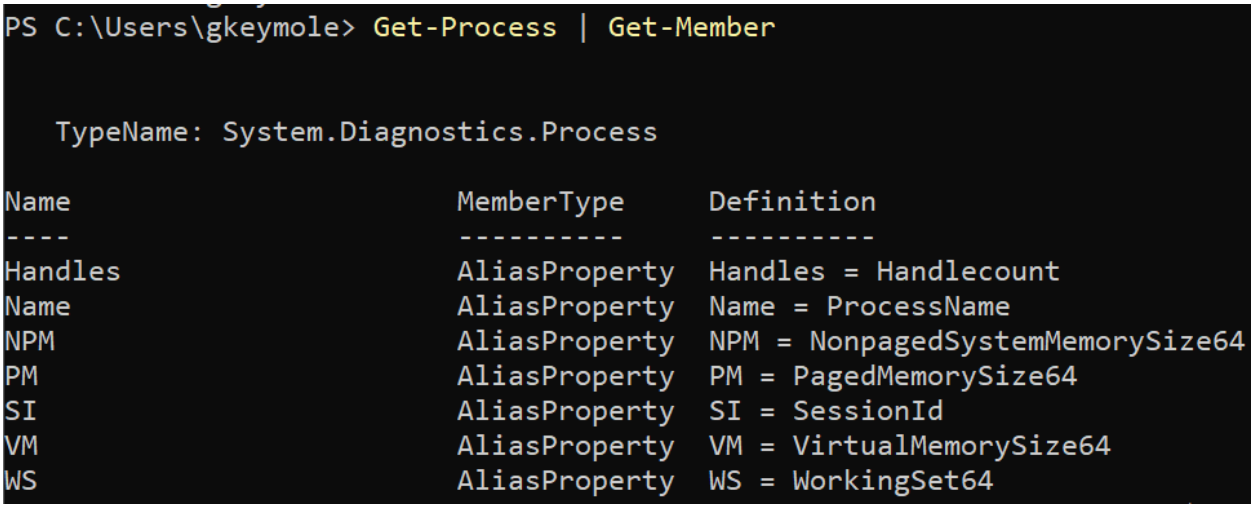
Command:

```
Get-Process | Get-Member
```

Explanation:

Displays all properties and methods for each process object in the pipeline.
Useful for identifying what info can be accessed or manipulated.

Screenshot:



```
PS C:\Users\gkeymole> Get-Process | Get-Member


   TypeName: System.Diagnostics.Process

Name                       MemberType     Definition
----                       ----------     ----------
Handles                    AliasProperty  Handles = Handlecount
Name                       AliasProperty  Name = ProcessName
NPM                        AliasProperty  NPM = NonpagedSystemMemorySize64
PM                         AliasProperty  PM = PagedMemorySize64
SI                         AliasProperty  SI = SessionId
VM                         AliasProperty  VM = VirtualMemorySize64
WS                         AliasProperty  WS = WorkingSet64
```

Task 2: Working with Objects

▨ Step 1: Display process information in table and list formats

Command:

```
Get-Process | Format-Table -AutoSize
Get-Process | Format-List
```

Explanation:

Shows process information in both tabular and detailed list formats. Table is compact; list is detailed with all fields.

Screenshots:

```
PS C:\Users\gkeymole> Get-Process | Format-Table -Autosize

Handles NPM(K)  PM(K)  WS(K) CPU(s)   Id SI ProcessName
------- ------  -----  ----- ------   -- -- -----------
    159     10   2524  11020   0.89 3024  0 AggregatorHost
     86      6   2452   5972   0.02 3680  1 cmd
    181     14   8120  28404  67.89 3696  1 conhost
    366     18   2012   6804   0.83  536  0 csrss
    208     15   1856   6568  28.92  640  1 csrss
    292     16   4136  17204   0.27 2748  0 dllhost
    638     32  28948  60924  79.27  844  1 dwm
     42      8   1960   5552   0.16  940  1 fontdrvhost
     42      7   1232   4284   0.02  948  0 fontdrvhost
      0      0     60      8           0  0 Idle
```

```
PS C:\Users\gkeymole> Get-Process | Format-List


Id      : 3024
Handles : 157
CPU     : 0.890625
SI      : 0
Name    : AggregatorHost

Id      : 3680
Handles : 86
CPU     : 0.015625
SI      : 1
Name    : cmd
```

▨ Step 2: Sort processes based on CPU usage

Command:

```
Get-Process | Sort-Object -Property CPU -Descending
```

Explanation:

Sorts processes by CPU usage in descending order to identify the most resource-consuming processes.

Screenshot:

```
PS C:\Users\gkeymole> Get-Process | Sort-Object -Property CPU -Descending

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1304       0       44        168      90.19       4   0 System
    692      94   263520     229740      80.88    1504   0 MsMpEng
    640      33    29964      61076      80.84     844   1 dwm
    181      14     8120      28404      69.48    3696   1 conhost
    976      82   185032     222960      55.73    2576   1 powershell
    378      27    11352      30292      41.28    3444   1 vmtoolsd
    208      15     1860       6572      29.28     640   1 csrss
    389      25     9980      27020      27.86    1112   0 vmtoolsd
    387      18     7884      20956      26.41    2052   0 svchost
```

▨ Step 3: Select and **filter** objects based on specific conditions

Command:

```
Get-Process | Where-Object { $_.CPU -gt 10 }
```

Explanation:

Filters and returns only the processes **using** more than 10 units of CPU time.

Screenshot:

```
PS C:\Users\gkeymole> Get-Process | Where-Object { $_.CPU -gt 10 }

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    181      14     8120      28404      70.03    3696   1 conhost
    208      15     1860       6568      29.55     640   1 csrss
    640      33    29964      61076      81.48     844   1 dwm
    690      94   263320     229652      80.98    1504   0 MsMpEng
   1027      82   185068     222996      56.13    2576   1 powershell
    365      17     7532      20800      26.56    2052   0 svchost
    257      16     8692      14320      10.75    4832   0 svchost
   1304       0       44        168      90.69       4   0 System
    139      10     1884       9044      11.16    2248   1 vm3dservice
    389      25     9940      27000      28.09    1112   0 vmtoolsd
    378      27    10068      29960      41.59    3444   1 vmtoolsd
    392      20    10228      26124      17.58    1624   0 WmiPrvSE
```

▨ Step 4: Loop through system objects and extract specific information

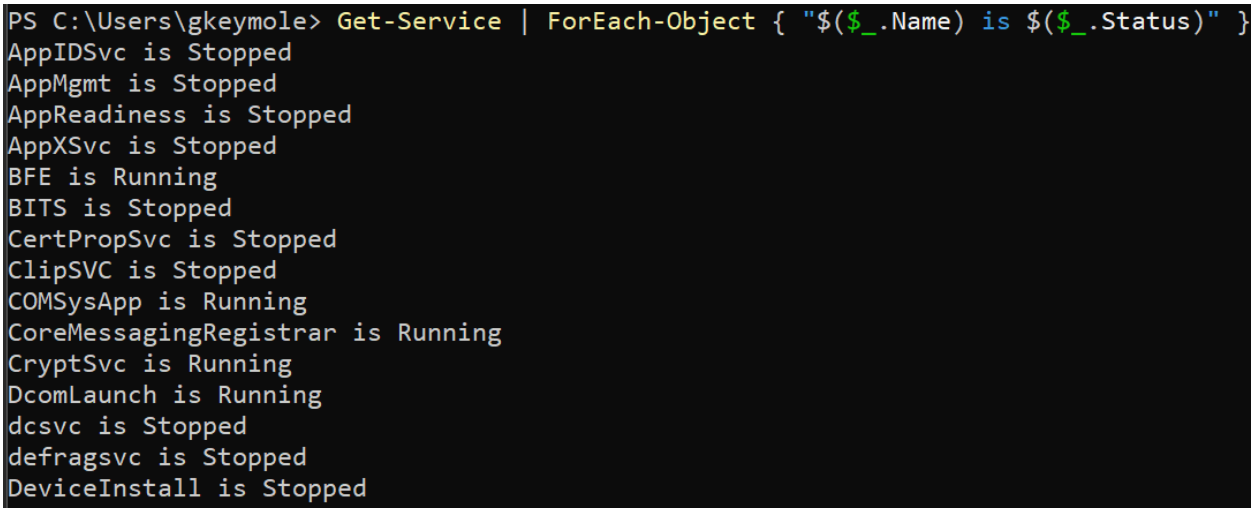
Command:

```
Get-Service | ForEach-Object { "$($_.Name) is $($_.Status)" }
```

Explanation:

Iterates through each service and prints its name and current status (Running, Stopped, etc.).

Screenshot:



Task 3: Managing the File System

▨ Step 1: Navigate through directories and list files

Command:

```
Get-ChildItem
Set-Location C:\Users
Get-Location
```

Explanation:

- `Get-ChildItem` lists files and folders in the current directory.
- `Set-Location` changes your working directory (like `cd`).
- `Get-Location` displays the current path.

Screenshots:

```
PS C:\Users\gkeymole> Get-ChildItem


    Directory: C:\Users\gkeymole


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---         4/30/2025   6:58 AM                Contacts
d-r---         4/30/2025   6:58 AM                Desktop
d-r---         4/30/2025   6:58 AM                Documents
d-r---         4/30/2025   6:58 AM                Downloads
d-r---         4/30/2025   6:58 AM                Favorites
d-r---         4/30/2025   6:58 AM                Links
d-r---         4/30/2025   6:58 AM                Music
d-r---         4/30/2025   6:58 AM                Pictures
d-r---         4/30/2025   6:58 AM                Saved Games
d-r---         4/30/2025   6:58 AM                Searches
d-r---         4/30/2025   6:58 AM                Videos
PS C:\Users\gkeymole> Set-Location C:\Users
PS C:\Users>_
PS C:\Users> Get-Location

Path
----
C:\Users
```

▨ Step 2: Create and delete files and folders

Command:

```
New-Item -Path "C:\TestFile.txt" -ItemType File
New-Item -Path "C:\TestFolder" -ItemType Directory
Remove-Item -Path "C:\TestFile.txt"
Remove-Item -Path "C:\TestFolder" -Recurse
```

Explanation:

- `New-Item` creates a new file or folder.
- `Remove-Item` deletes files or folders; `-Recurse` ensures folder contents are also removed.

Screenshots:

```
PS C:\> New-Item -Path "C:\TestFile.txt" -ItemType File


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/30/2025  11:25 AM              0 TestFile.txt
PS C:\> New-Item -Path "C:\TestFolder" -ItemType Directory


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/30/2025  11:26 AM                TestFolder
PS C:\> ls


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/30/2025   8:30 AM                Folder1
d-----          4/1/2024   1:01 AM                PerfLogs
d-----         4/30/2025   8:53 AM                PowershellLab
d-r---         4/30/2025   6:58 AM                Program Files
d-r---          4/1/2024   1:11 AM                Program Files (x86)
d-----         4/30/2025  11:26 AM                TestFolder
d-r---         4/30/2025   6:58 AM                Users
d-----         4/30/2025   7:47 AM                Windows
d-----         4/30/2025   8:56 AM                Windows.old
-a----         4/30/2025  11:25 AM              0 TestFile.txt
PS C:\> Remove-Item -Path "C:\TestFile.txt"
PS C:\> Remove-Item -Path "C:\TestFolder" -Recurse
```

▨ Step 3: Copy and move files between locations

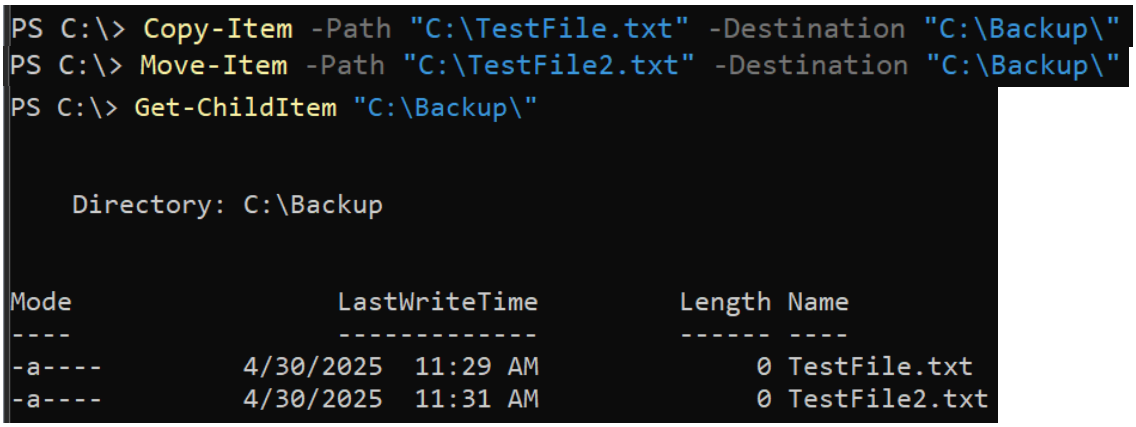
Command:

```
Copy-Item -Path "C:\TestFile.txt" -Destination "C:\Backup\"
Move-Item -Path "C:\TestFile.txt" -Destination "C:\Backup\"
```

Explanation:

- `Copy-Item` duplicates the file to another folder.
- `Move-Item` moves the file to a new location.

Screenshot:



▨ Step 4: Check available disk space

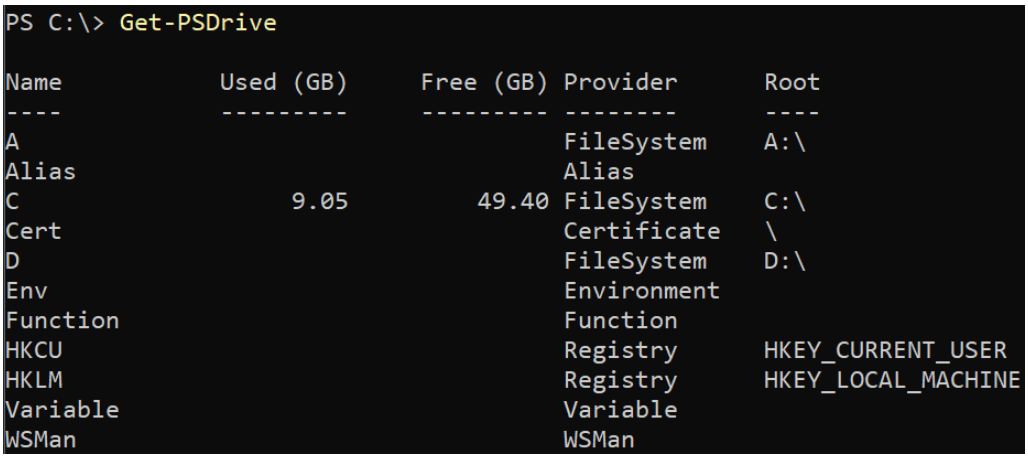
Command:

```
Get-PSDrive
```

Explanation:

Displays all logical drives, including their free space and used space.
Helpful **for** monitoring storage capacity.

Screenshot:



Task 4: Managing System Services and Processes

▨ Step 1: List all running services on the system

Command:

```
Get-Service | Where-Object { $_.Status -eq 'Running' }
```

Explanation:

Displays only the services that are currently running. Useful for system monitoring and troubleshooting.

Screenshot:

```
PS C:\Users\gkeymole> Get-Service | Where-Object { $_.Status -eq 'Running' }

Status   Name               DisplayName
------   ----               -----------
Running  BFE                Base Filtering Engine
Running  COMSysApp          COM+ System Application
Running  CoreMessagingRe... CoreMessaging
Running  CryptSvc           Cryptographic Services
Running  DcomLaunch         DCOM Server Process Launcher
Running  Dhcp               DHCP Client
Running  DiagTrack          Connected User Experiences and Tele...
Running  DispBrokerDeskt... Display Policy Service
Running  Dnscache           DNS Client
Running  DPS                Diagnostic Policy Service
Running  EventLog           Windows Event Log
Running  EventSystem        COM+ Event System
Running  gpsvc              Group Policy Client
```

▨ Step 2: Start and stop specific services

Command:

```
Start-Service -Name "wuauserv"
Stop-Service -Name "Spooler"
```

Explanation:

- `Start-Service` starts the Windows Update service.
- `Stop-Service` stops the Print Spooler service.

Screenshot:

```
PS C:\Users\gkeymole> Start-Service -Name "wuauserv"
PS C:\Users\gkeymole> Stop-Service -Name "Spooler"
```

▨ Step 3: Retrieve information about active processes

Command:

```
Get-Process
```

Explanation:

Lists all currently running processes with details like CPU time, memory usage, and process ID.

Screenshot:

```
PS C:\Users\gkeymole> Get-Process

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
    157       9     2460      10984       0.91    3024   0 AggregatorHost
     86       6     2452       5972       0.02    3680   1 cmd
    181      14     5952      26132      79.33    3696   1 conhost
    366      18     2016       6812       0.86     536   0 csrss
    208      15     1860       6572      33.27     640   1 csrss
    292      16     4136      17204       0.28    2748   0 dllhost
    638      33    30004      61112      96.00     844   1 dwm
     42       8     1960       5552       0.16     940   1 fontdrvhost
     42       7     1232       4284       0.02     948   0 fontdrvhost
      0       0       60          8                  0   0 Idle
    952      22     5216      20036       3.86     784   0 lsass
    398      21     9184      32504       4.31    3324   0 MoUsoCoreWorker
```

▨ Step 4: Terminate a process

Command:

```
Stop-Process -Name "notepad"
# or by ID:
Stop-Process -Id 1234
```

Explanation:

Terminates a running process either by name (e.g., notepad) or by specifying its process ID.

Screenshot:

```
PS C:\Users\gkeymole> Stop-Process -Name "notepad"
```

```
PS C:\Users\gkeymole> Stop-Process -Id 1234
```

Task 5: Monitoring Event Logs and System Information

▨ Step 1: View the latest system event logs

Command:

```
Get-EventLog -LogName System -Newest 10
```

Explanation:

Retrieves the 10 most recent entries from the System event log. Useful for diagnosing system-level issues.

Screenshot:

```
PS C:\Users\gkeymole> Get-EventLog -LogName System -Newest 10

   Index Time          EntryType   Source                 InstanceID Message
   ----- ----          ---------   ------                 ---------- -------
     622 Apr 30 11:42  Information Service Control M...   1073748860 The Windows Update service entered...
     621 Apr 30 11:34  Warning     Microsoft-Windows...         1014 Name resolution for the name wpad ...
     620 Apr 30 11:25  Information Service Control M...   1073748860 The AppX Deployment Service (AppXS...
     619 Apr 30 11:24  Warning     Microsoft-Windows...         1014 Name resolution for the name wpad ...
     618 Apr 30 11:20  Information Service Control M...   1073748860 The Software Protection service en...
     617 Apr 30 11:20  Information Service Control M...   1073748860 The AppX Deployment Service (AppXS...
     616 Apr 30 11:20  Information Service Control M...   1073748860 The Software Protection service en...
     615 Apr 30 11:18  Information Service Control M...   1073748860 The Windows Update service entered...
     614 Apr 30 11:08  Information Service Control M...   1073748860 The WaaSMedicSvc service entered t...
     613 Apr 30 11:07  Information Service Control M...   1073748860 The WaaSMedicSvc service entered t...
```

▨ Step 2: Retrieve security event logs

Command:

```
Get-WinEvent -LogName Security
```

Explanation:

Displays entries from the Security event log. These include login attempts, account changes, and other security-related events.

Screenshot:

```
PS C:\Users\gkeymole> Get-WinEvent -LogName Security


   ProviderName: Microsoft-Windows-Security-Auditing

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
4/30/2025 11:42:52 AM         4672 Information      Special privileges assigned to new logon....
4/30/2025 11:42:52 AM         4624 Information      An account was successfully logged on....
4/30/2025 11:20:10 AM         4672 Information      Special privileges assigned to new logon....
4/30/2025 11:20:10 AM         4624 Information      An account was successfully logged on....
4/30/2025 11:07:42 AM         4672 Information      Special privileges assigned to new logon....
4/30/2025 11:07:42 AM         4624 Information      An account was successfully logged on....
4/30/2025 11:07:40 AM         4672 Information      Special privileges assigned to new logon....
4/30/2025 11:07:40 AM         4624 Information      An account was successfully logged on....
4/30/2025 11:01:36 AM         4672 Information      Special privileges assigned to new logon....
4/30/2025 11:01:36 AM         4624 Information      An account was successfully logged on....
```

▨ Step 3: Extract operating system details using PowerShell

Command:

```
Get-WmiObject -Class Win32_OperatingSystem
```

Explanation:

Provides detailed OS information such as version, build number, architecture, and install date.

Screenshot: