# Lab 1: **Installation of PowerShell 7**

1. Refer given link for installation of PowerShell 7 on Windows

[Installing PowerShell on Windows - PowerShell | Microsoft Learn](https://learn.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.3&viewFallbackFrom=powershell-7)

# Lab 2: **Execution of basic cmdlets**

1. Get list of commands available with PowerShell

Get-Command

1. Get Help for a command ‘Write-Warning’

Get-Help Write-Warning

1. Get list of verbs used by most of the commands

Get-Verb

1. Get list of objects associated with command ‘Get-Process’

Get-Process | Get-Member

Get-Verb | Get-Member

# Lab 3: **File I/O using cmdlets**

1. Create a folder

New-Item C:\test -ItemType Directory

1. Create a file named try.txt

New-Item C:\test\try.txt

1. Get files present in a folder

Get-ChildItem C:\test

1. Add content to the file

Set-Content C:\test\try.txt 'Welcome to Data Preparation session'

1. Read content from the file

Get-Content C:\test\try.txt

1. Add more contents to the file

Add-content C:\test\try.txt 'Enjoy working with PowerShell'

Get-content C:\test\try.txt

1. Rename folder and file

Rename-Item C:\test test1

Rename-Item C:\test1\try.txt try1

1. Erase contents from a file

Clear-Content C:\test1\try1.txt

Get-content C:\test1\try1.txt

1. Delete a file

Remove-Item C:\test1\try1.txt

Get-content C:\test1\try1.txt

1. Delete folder

Remove-Item C:\test1

# Lab 4: **Scripting in PowerShell**

1. Using variable get value for an individual object

* Create variable and assign it with value of current path

$loc = Get-Location

* Get variable value

$loc

* Get list of objects from a variable

$loc | Get-Member

* Get a specific object from a variable

$loc | Select-Object {$\_.Path}

1. Write a script to find out which files have been modified since 1st July 2023

* Get child items from path C:\

Get-ChildItem -Path C:\

* Get list of objects for ChildItem

Get-ChildItem | Get-Member

* Using an object LastWriteTime find out all files modified since 1st July

Get-ChildItem -Path C:\ | Where-Object {$\_.LastWriteTime -gt "07/01/2023"}

# Lab 5: **Declare 2 variables and compare values**

$a = 5

$b = 6

if($a -eq $b)

>> {"Equal"}

>> else

>> {"No Equal"}

# Lab 6: **Find out list of processes which start with ‘Micro’**

>> foreach($Process in Get-Process)

>> {

>> if($Process.Name -like "Micro\*")

>> {

>> $Process.Name

>> }

>> }

# Lab 7: **Displaying numbers from 1 to 10**

>> for($x=1; $x -lt 10; $x=$x+1)

>> {

>> $x

>> }

# Lab 8: **To recursively check notepad application is running or not**

Initially open the notepad. Execute the script. After a while close the notepad application.

$Proc = Invoke-Command {Get-Process -Name notepad}

>> while($Proc){

>> $Proc = Invoke-Command {Get-Process -Name notepad}

>> Write-Host "Running"

>> Start-Sleep -Seconds 5

>> }

>> Write-Host "Not Running"

# Lab 9: **Guess Lucky Number**

>> $lucky\_num=91

>> do

>> {

>> $guess = Read-Host -Prompt "Guess 2-digit number : "

>> } while($guess -ne $lucky\_num)

>> Write-Host "Congratulations.. You are lucky"

>>

# Lab 10: **Write PowerShell script in a file and then execute it**

* Create a file ‘first.ps1’. Make sure extension of file must be .ps1 not .txt
* Add given code to the file

$today = Get-Date

$yesterday = $today.AddDays(-1)

$tomorrow = $today.AddDays(1)

Write-Host "Today: `t`t$today"

Write-Host "Yesterday: `t$yesterday"

Write-Host "Tomorrow: `t$tomorrow"

Start-Process wordpad.exe

exit 0

* On PowerShell, execute command ./first.ps1.