PowerShell terminal:

Notepad.exe

# Call a method to stop the Notepad.exe by using ForEach-Object command:

Get-Process Notepad | ForEach-Object Modules

# Enumerate only the string values of all service names by using ForEach-Object command:

Get-Service | ForEach-Object Name

# Output the same list of strings to a file named C:\Class\AllServiceNames.txt

Get-Service | ForEach-Object Name | Out-File –FilePath C:\Class\AllServiceNames.txt

# Retrieve all Running services that contain “Win” or “Microsoft”

# Use ForEach-Object to extract only the string values of the service names

# Output the resulting strings to a file named C:\Class\ServiceTemplate.txt

Get-Service | Where-Object {$\_.Status –eq “Running” –and ($\_.DisplayName –like “\*win\*” –or $\_.DisplayName –like “\*Microsoft\*”)} | ForEach-Object Name | Out-File –FilePath C:\Class\ServiceTemplate.txt

# Call the Stop method for the FontCache, W32Time and WinRM services by using the ForEach-Object command:

Get-Service FontCache,W32Time,WinRM | ForEach-Object Stop

# Retrieve the contents of the C:\Class\ServiceTemplate.txt

# Find the current service objects that match the names from the text file

# Determine which of those services are currently Stopped

Gc C:\Class\ServiceTemplate.txt | gsv | ? Status –eq Stopped | %Start

# Start a Notepad.exe

Notepad.exe

# Expand the list of Modules associated with Notepad.exe by using ForEach-Object:

Get-Process Notepad | ForEach-Object {$\_.Modules}

# Call a method to stop Notepad.exe by using ForEach-Object:

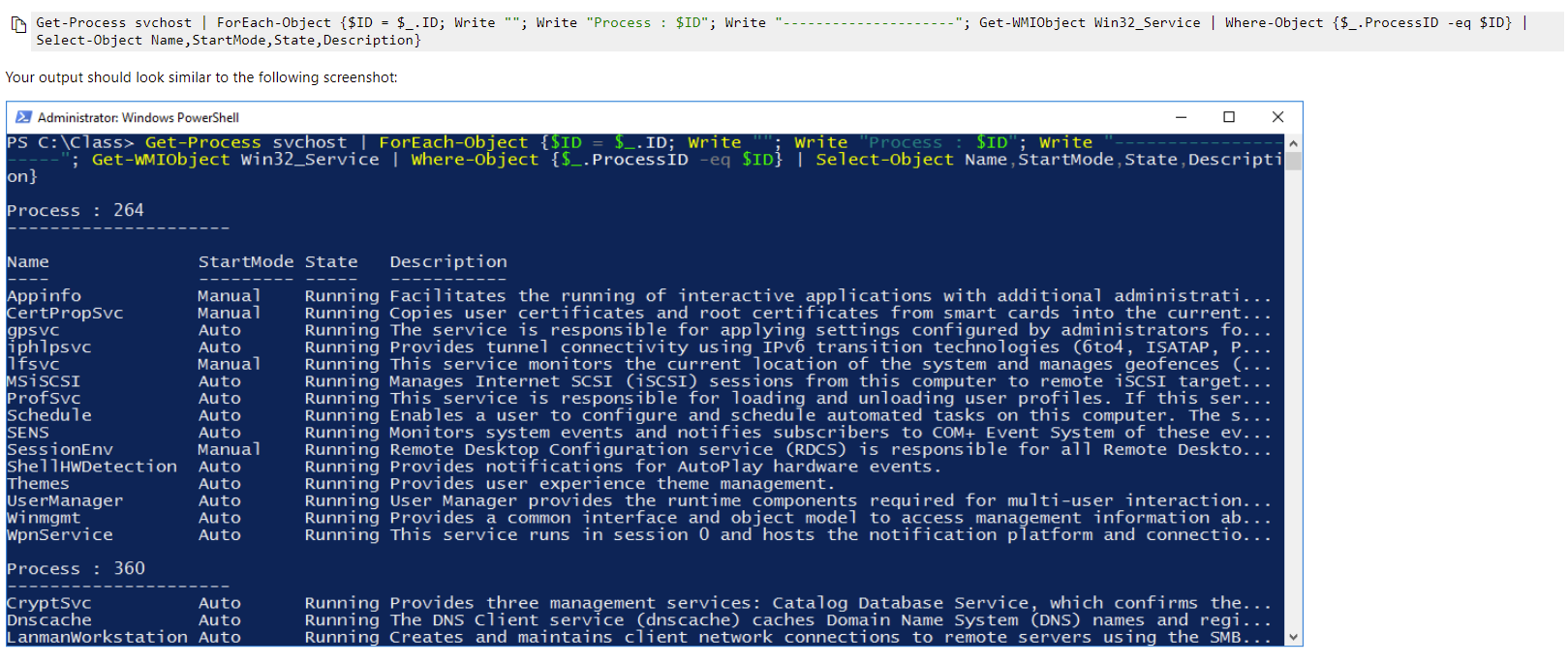
Get-Process Notepad | ForEach-Object {$\_.Kill()}

# Retrieve a list of all SVCHost processes

# Use ForEach-Object to find services that have the same ProcessID as the current SVCHost in the pipeline

# Output the resulting list with the process ID at the top of each list, followed by the services that match the process ID

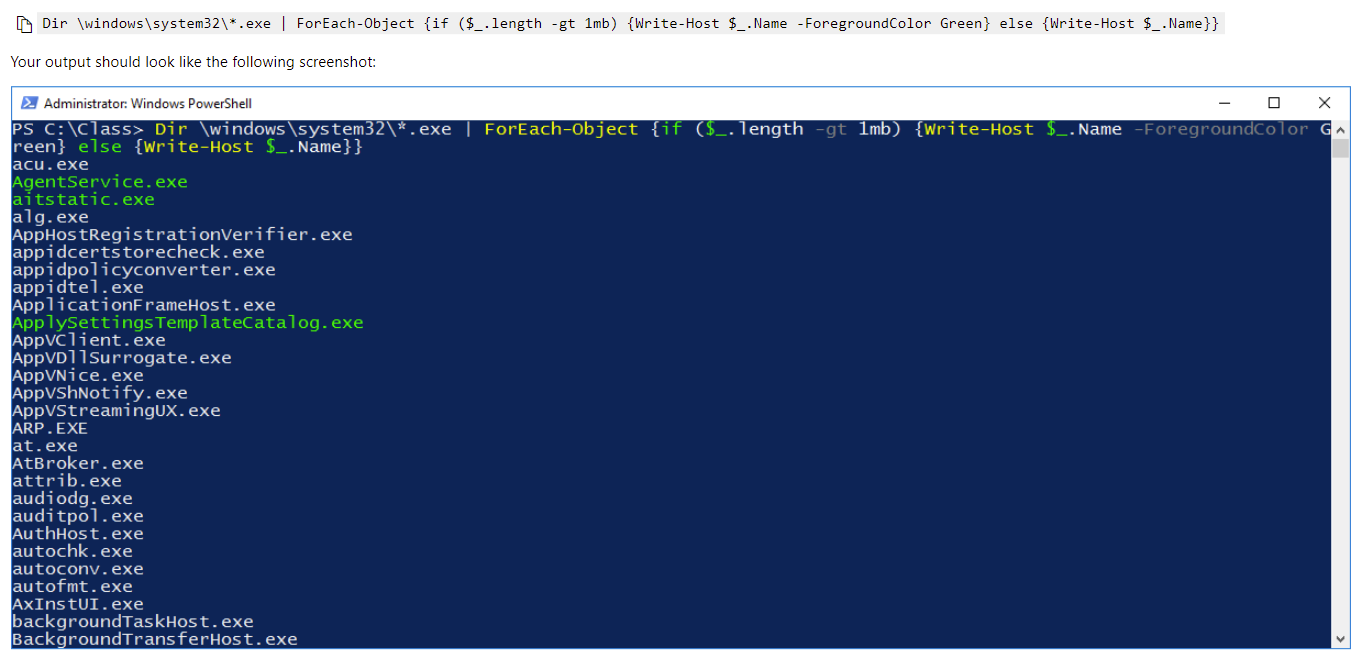
# Display the processes using the Name, StartMode, State, Description properties as headings

Get-Process svchost | ForEach-Object {$ID = $\_.ID; Write “”; Write “Process : $ID”; Write “------------“; Get-WMIObject Win32\_Service | Where-Object {$\_.ProcessID –eq $ID} | Select-Object Name,StartMode,State,Description}

# Retrieve a list of EXE files from C:\Windows\System32

# Use ForEach-Object to display the names of files larger than 1 megabyte in green

# Display all other file names using the default colors

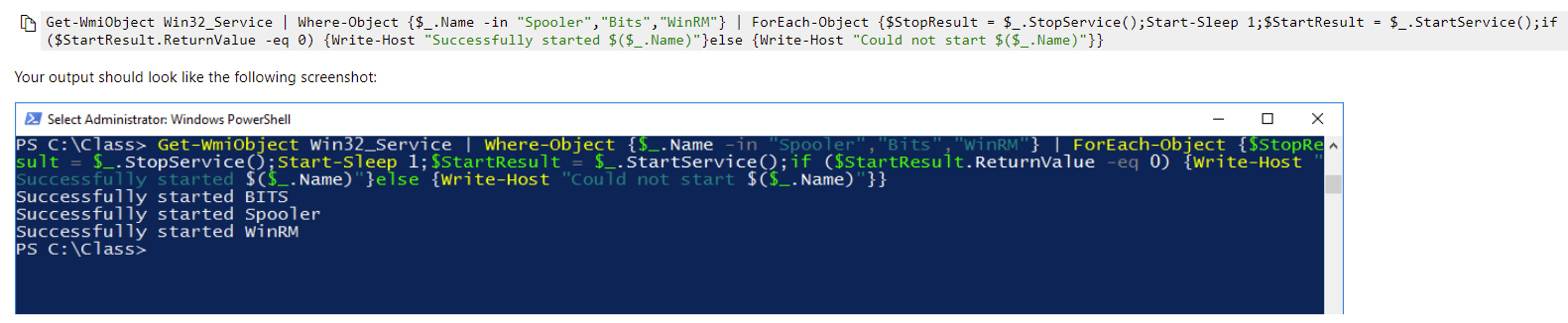
Dir C:\windows\system32\\*.exe | ForEach-Object {if ($\_.length –gt 1mb) {Write-Host $\_.Name –ForegroundColor Green} else {Write-Host $\_.Name}}

# Retrieve the Spooler, Bits, and WinRM services using a WMI query

# Use ForEach-Object to call the StopService and StartService methods of the services

# Insert a one second pause between the StopService and StartService methods

# Capture the result of the StopService and StartService commands, and then write “Successfully started” or “Cloud not start”, along with the service name, to the screen based on the result:

Get-WmiObject Win32\_Service | Where-Object {$\_.Name –in “Spooler”,”Bits”,”WinRM”} | ForEach-Object {$StopResult = $\_.StopService();Start-Sleep 1;$StartResult = $\_.StartService();if $StartResult.ReturnValue –eq 0) {Write-Host “Successfully started $($\_.Name)”} else {Write-Host “Could not start $($\_.Name)”}}