**PowerShell Tasks**

**For each of these tasks, take screenshots, and ensure that your name is on the screenshot.  
One of the best ways to do this is to create a directory which is called your name, and thus your name will be in the prompt.**

**Task 1** : Discover the necessary commands for working with installed hotfixes, and use **Select-Object** and **Sort-Object** to customize the final output of each command as specified.

a)  Using a keyword like **hotfix**, find a command that can display a list of installed hotfixes.

b)  Display the members of the object produced by the command that you found in the previous step.

c)  Display a list of installed hotfixes. Display only the installation date, the hotfix ID number, and the name of the user who installed the hotfix.

**Task 2 : Display a Sorted List of Enabled Windows Firewall Rules**

In this task, you will discover the necessary commands for working with firewall rules, and use **Select-Object** and **Sort-Object** to customize the final output of each command as specified.

a)  Using a keyword such as **rule**, find a command that can display firewall rules.

b)  Display a list of firewall rules.

c)  View the Help for the command that displays firewall rules.

d)  Display a list of firewall rules that are enabled. : Display a list of enabled firewall rules. Display only the rules’ display names, the profile they belong to, their direction, and whether they allow or deny access. Sort the list in alphabetical order by display name. **Hint**: Use **Get-Member** to determine the correct property names.

**Task 3 | Filter Security Event Log Entries**

In this task, you will create a report that shows Security event log entries having the event ID 4624.

Note: You may need to adjust this lab for your computer. You many need to use a different event log or select criteria.

a) Display a list of Security event log entries that have the event ID 4624.  
b) Display the list again and show only the time written, event ID, and message.

c) Produce the same list in an HTML file named EventReport.html.  
d) View EventReport.html.

**Task 4 | Filter Encryption Certificates**

In this task, you will display a list of encryption certificates installed on the computer. Note: The CERT: drive used in this task is a PSProvider.

a)  Display a directory listing of all items in the CERT: drive. Include subfolders in the list.

b)  Display the members of the objects produced by the directory listing.

c)  Display the list again and show only certificates that do not have a private key.

d)  Display the list again and show only certificates that have a NotBefore date that is before today, and a NotAfter date that is after today. Include only certificates that do not have a private key.

e)  Display the list again and show only the issuer name, NotAfter date, and NotBefore date for each certificate.

**Task 5 | Filter Disk Volumes**

In this task, you will create a report that shows disk volumes that are running low on space.

a)  Display a list of disk volumes.

b)  Display a list that shows the members of the objects produced by the previous command.

c)  Display a list of volumes that have more than zero bytes of free space.

d)  Display a list of volumes that have less than 99 percent free space, and more than zero bytes of free space.

e)  Display a list of volumes that have less than 10 percent free space and more than zero bytes of free space. This command may produce no results if no volumes on your computer meet the criteria.

**Task 6 | Filter Control Panel Items**

In this task, you will create a report that displays specified Control Panel items.

a)  Using a keyword like control, find a command that can display Control Panel items.

b)  Display a list of all Control Panel items.

c)  Display a list of Control Panel items in the System and Security category.

**Task 7 | Create a Simple Table**

1. a)  Use Get-Process to display the running processes.
2. b)  Create an output table that includes: Process Name, ID, Virtual Memory, and Physical Memory. The table should not have additional space between columns.

**Task 8 | Create a Customized Table**

In this task, you will create a table of TCP/IP routes.

To display a list of local TCP/IP routes run: Get-NetRoute.  
Create an output table that includes: Route Address Family, Route Metric, Type of Route, and Destination Prefix.  
The table should not have additional space between columns.  
Information should be grouped by Route Address Family.  
Table header should not display.

**Task 9 | Display File Names and File Sizes**

In this task, you will customize the display of files and file sizes.  
Use Get-ChildItem (not an alias for it).  
Hint: In this task you are going to to create a sequence of commands. Build and test each command before joining them together.

a)  Write a command that will display a list of all the c:\Windows files with an .exe file name extension.

b)  Sort the output of the previous command by the file size.

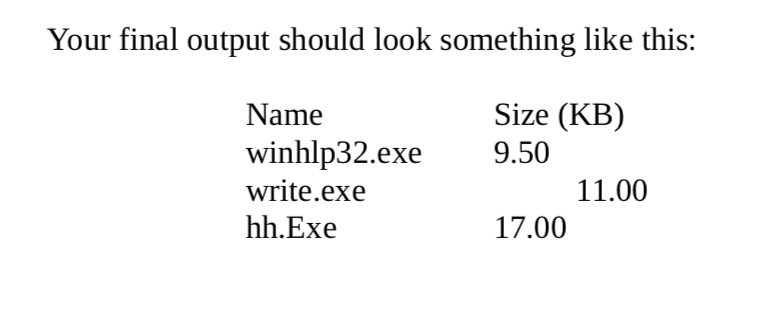
c)  Provide tabular output of the previous command that includes only Name and

Length. The table should be automatically size to fit the display.

d)  Change the name of the Length column to "Size (KB)".

e)  Convert the length property to kilobytes.

f)  Provide only two decimal places on the length calculation.



**Task 10 | Display Event Logs**

In this task, you will customize the display of the Security event log. Specifically, we will determine if there is a delay between when an event is generated and when it is written to the event log.  
Note: Don't be surprised if there aren't any time delays. That might be a good thing!

a)  Display the Security event log.

b)  Display only the newest 10 entries in the Security event log.

c)  Use Get-Member to determine the correct property names for Event ID

Number, Time Generated, and Time Written to the Log.

d)  Use Select-Object to display only the previous three properties.

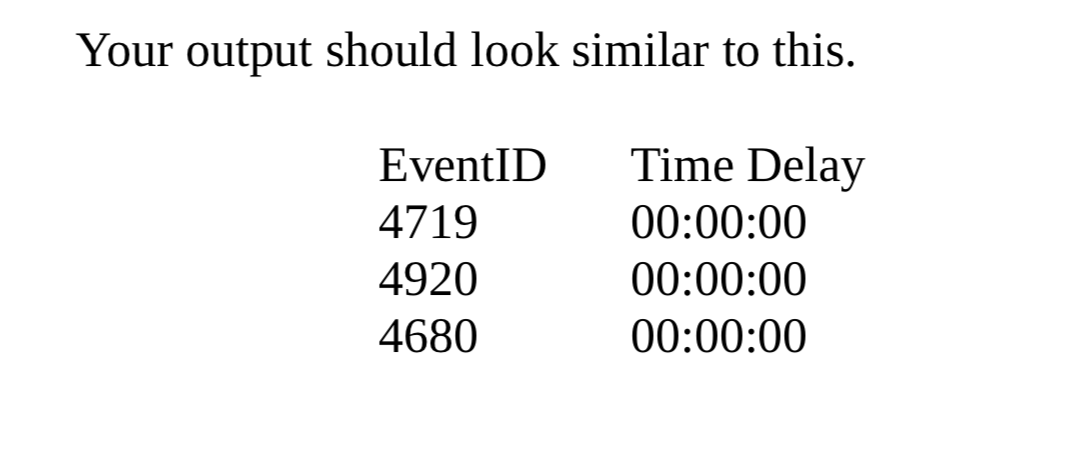
e)  Use Select-Object to display another property called "Time Delay". This will

be a calculated property displaying the difference between Time Written and

Time Generated.

f)  Use Sort-Object to sort by Time Delay.

g)  Use Format-Table to display only the EventID and Time Delay.Autosize the table.



**Task 11 | Using Out-GridView**

Look up this cmdlet.  
In this task, you will use Out-GridView to customize the Get-Process output.

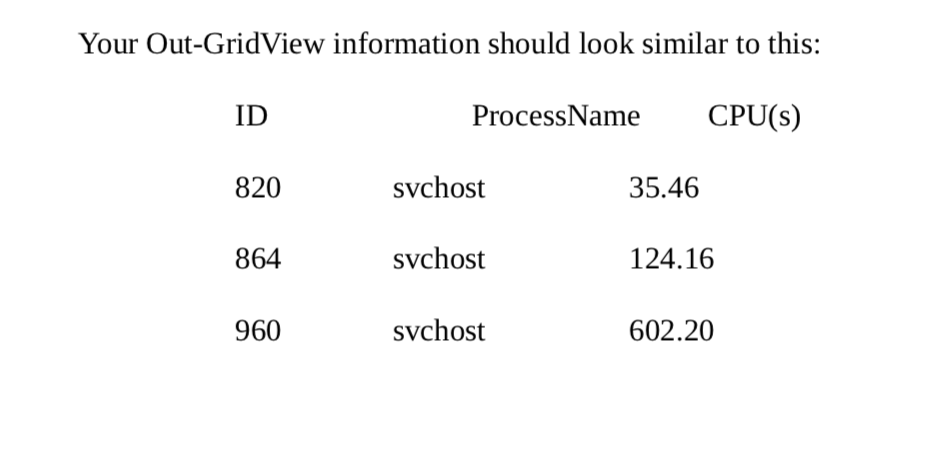
a)  Pipe the Get-Process command to Out-GridView. Set the Title of the Gridview to "Processes by CPU".

b)  Display only the Event ID, Process Name, and CPU properties; in that order.

c)  Display only processes named svchost.

d)  Sort the display by Event ID.

e)  Copy the information to Notepad.



**Task 12 | Samll script that reads/writes a file, and uses the datetime type**

Use **systeminfo** in a script which calculates how long this system has been up in minutes and seconds.  
You must use systeminfo, and send its output to a temporary file file.  
Then, as part of the script, read from that file, get the boot time, and compare with the time now.

Don’t use e.g. get-wmiobject to achieve this task.