**TPG527C**

Project 2:

PowerShell basics

Object, Properties, Methods

Input and Output

Scripting

Statements

Due date: 22 August 2023

Restrictions:

* This is not a group project. You may not consult with anybody. It is you and your knowledge only. Read the details on the declaration form properly. Under no circumstances will I allow collaboration of any kind.

* You may not use ChatGPT or any other AI tool.

* You may use a search engine such as Google, but you must reference the exact url where you got the answer. No reference, no marks.

Preparation

Follow the steps in the tutorial and create a Virtual machine.

Assignment

The following questions pertain to Windows PowerShell. Use the PS command window for all the questions.

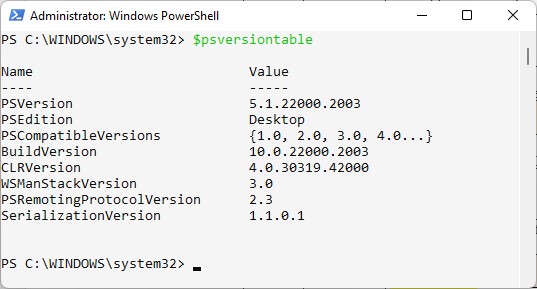
For Question 1 - 9:

* For each question, list all steps and provide screen prints of each intermediate step as it is on your computer.
* Use Alt-PrintScreen, Gadwin PrintScreen or Snipping Tool to make screen prints of the relevant parts of a screen. Please do not use PrintScreen on its own. I do not want to see your whole screen with the background content.

**1. What version of PowerShell are you using?**

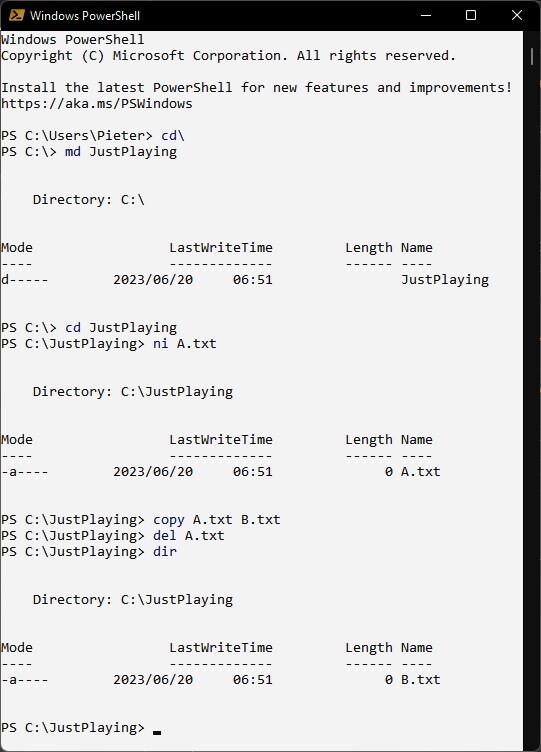
Make a screen print of your PS command window to show the command that you used along with its output.

$psveriontable



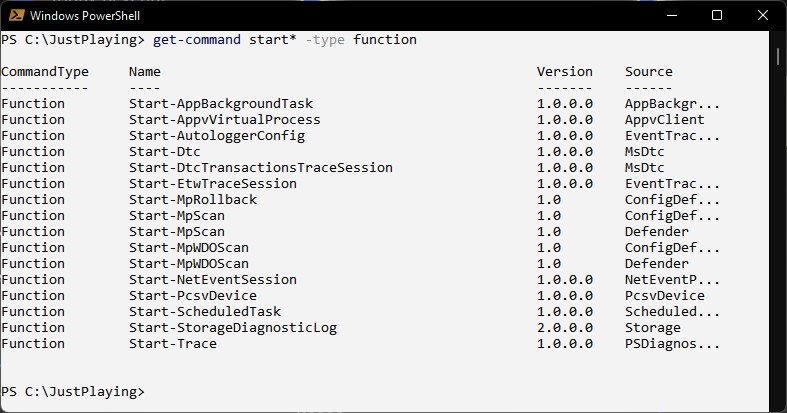
# Folders and files

* Create a folder on C, namely C:\JustPlaying.
* Create a text file, A.txt, in the folder JustPlaying.
* Copy A.txt to B.txt
* Delete A.txt
* List all files in C:\JustPlaying



# List all cmdlet functions that start with "start-"

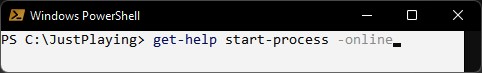
Get-command start\* -type function



# Get online help

Get online help on the start-process command.

Get-help start-process -online

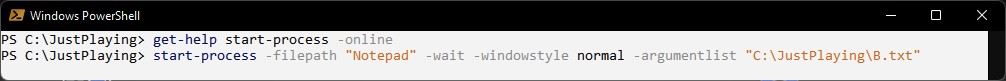


# Start a process

Use the help in the previous question to start Notepad in a normal window and open the file B.txt that was created above. It should not be possible to do anything in the PS Window before Notepad is closed again.

>Get-help start-process -online

>start-process -filepath “Notepad” -wait -windowstyle normal -argumentlist “c:\JustPlaying\B.txt”

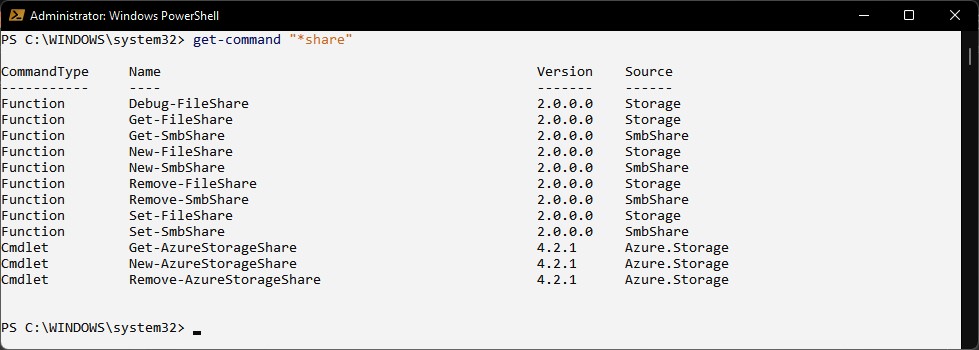


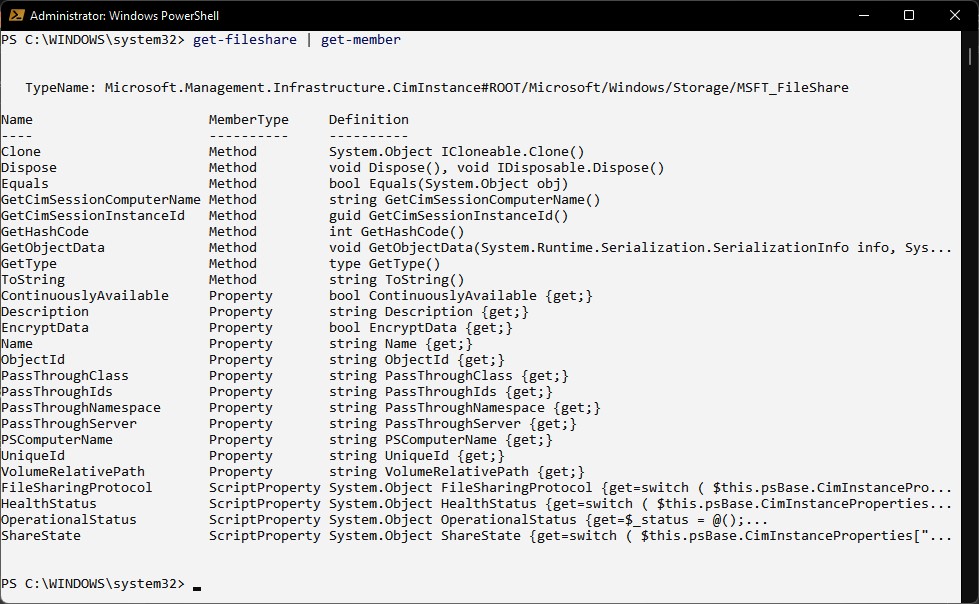
# List file shares

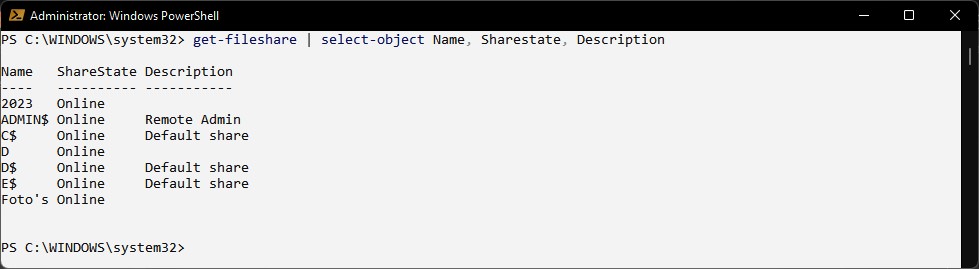
Use the help facilities to figure out how to list the name, share state and description of all file share objects on the computer.

Provide three screen prints:

* The command and results to determine the command to use. In other words, how did you know which command will list the file shares?
* The command and result to determine the available members of the command. How did you know what members are available and how to spell them?
* The command and results of the final command to list the shares.
* > get-command “\*share”  
  >get-fileshare | get-member  
  >get-fileshare | select-object Name, Sharestate, Description



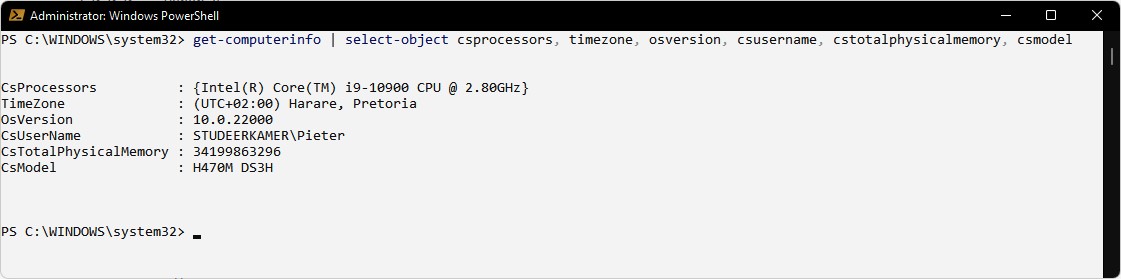




# List the following details of your computer

Processor details, motherboard model, username, OS version, time zone

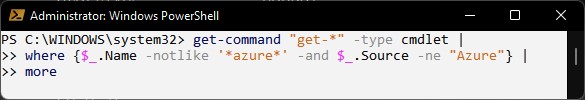
>get-computerinfo | select-object csprocessors, timezone, osversion, csusername, cstotalphysicalmemory, csmodel



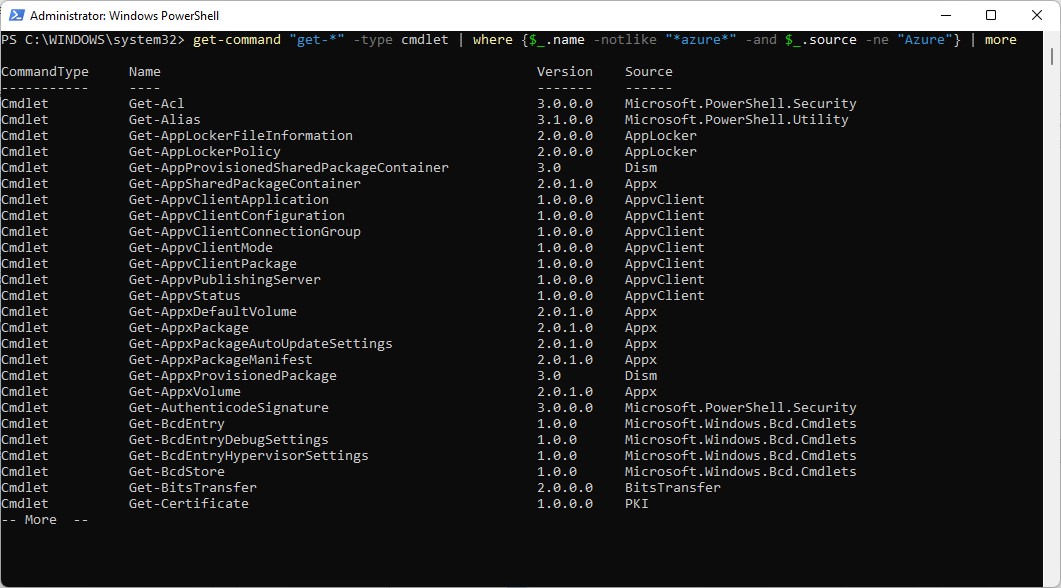
# List cmdlets

List all cmdlets that start with "get-" and does not contain "azure". The source should also not be Azure. Pause the output after a page.

>get-command “get-\*” -type cmdlet | where [$\_.Name -notlike ‘\*azure\*’ -and $\_.Source -ne “Azure”} | more



(Use Shift-Enter to break at natural line break characters)



# Enter data into text file

* Make sure that the file, C:\JustPlaying\B.txt, is empty.
* Use a PS command line to enter your name as hard-coded string into the file.
* Read your age from the keyboard and add it to the file. - Display the content of the file on the screen.

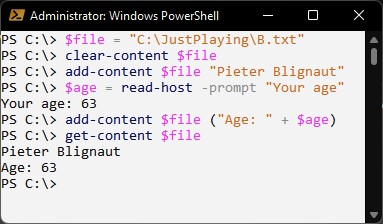
My file looks like this:



>$file = “C:\JustPlaying\.txt”  
 >clear-content $file  
 >add-content $file “Mpho Wittes”  
 >$age = read-host -prompt “Your age”  
 Your age: 20

>add-content $file (“Age: “ + $age)

>get-content $file  
 >20



For Question 10 and 11:

This will be marked negatively - if these instructions are not adhered to, marks will be subtracted.

* Google is your friend as long as all urls are properly referenced in a comment block at the end of the script.
* Your name and student number must appear in a comment block at the top of the script.
* Provide comment lines above every line of code to explain what you are doing.
* Make sure that you clear the console at the beginning of every script.
* Make sure that your code is properly indented and that matching braces are aligned. Readability is key!!

# File properties

A few text files with random questions to ChatGPT and its answers are provided.

*(By the way, you may never ever use ChatGPT or any other AI tool to assist you with a project!! If I catch you, I will ask you friendly to please leave the course.)*

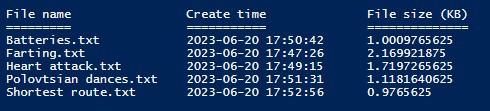
Copy the files to a dedicated folder on your PC.

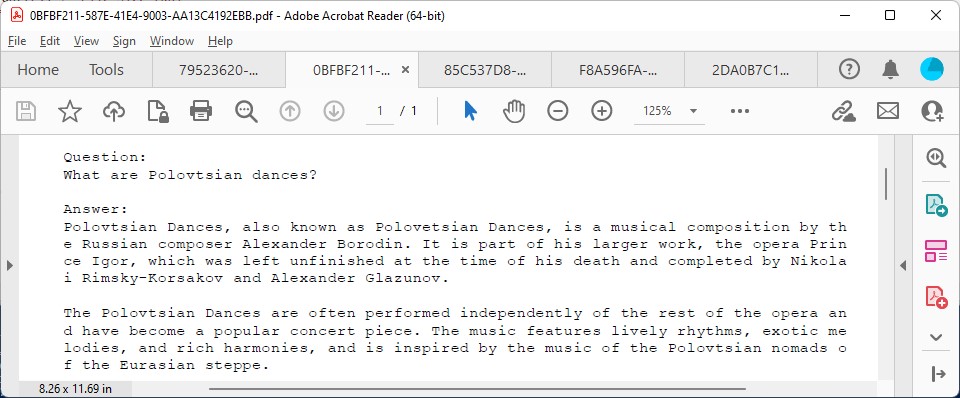
Develop a PS script to step through all files in the folder. List the file name, date of creation, and file size. Print the content of each file to a pdf printer

Hints

* See also Sections 10.1.1 and 10.1.7 in the tutorial document.
* Download the Bullzip pdf printer. It has a property to create a file name based on a system generated GUID so that you don’t have to enter a file name for every file individually.

This is my output. Note that the columns are aligned and that there are headers. Note also that the create time is neatly formatted.





#Clear console

cls

#Variables for folder and files

$folder = 'D:\Dropbox\CUT\Operating systems\2023\Projects\Project 2\Practice files'

$files = dir -path $folder

#Header

"`n"

"File name".PadRight(25) + "Create time".PadRight(25) + "File size (KB)"

"=====================".PadRight(25) + "===================".PadRight(25) + "=============="

#Step through files

foreach ($file in $files)

{

# Formatted create date

$cdate = ($file.CreationTime).ToString("yyy-MM-dd HH:mm:ss")

#Output

#$file.FullName

$file.name.PadRight(25) + $cdate.PadRight(25) + ($file.Length/1KB).ToString().PadRight(15)

#Read content and print

$content = Get-Content $file.FullName

$content | out-printer -name "Bullzip PDF Printer" #Alternative "Microsoft Print to PDF" - will ask for file name everytime

#Note: The Bullzip PDF printer has a seting to automatically generate a file name based on a system-generated GUID

}

<# References

https://shellgeek.com/powershell-get-file-size/

https://stackoverflow.com/questions/65760455/how-to-automate-print-to-pdf-with-windows-powershell

#>

# Numbers

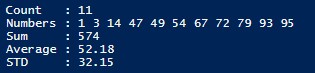
Create an array of random integer values.

Ask the user how many numbers there must be in the array.

List the numbers in numerical order.

Display the sum, average and standard deviation of the numbers.

Example outputs:







Notes:

* The numbers are sorted in numerical order and listed on a single line - The average and standard deviation are rounded to 2 decimal digits.
* Make sure that you do not get a runtime error if the user enters 0 or an invalid numerical value for the number of numbers.

Hints

* [https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/variancestandard-deviation-population/a/calculating-standard-deviation-step-by-step](https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/variance-standard-deviation-population/a/calculating-standard-deviation-step-by-step)

#Clear console

cls

"`n"

#Create array

$numbers = @()

Try

{

#Get number of numbers from user

[int] $n = Read-Host -prompt "Count "

#Add random numbers to the array

for ($i=1; $i -le $n; $i++)

{

$numbers += Get-Random -Minimum 1 -Maximum 100

} #for i

#Initialise stats

$sum = 0

#Step through array

write-host "Numbers : " -NoNewline

foreach ($number in $numbers | Sort-Object)

{

Write-Host $number '' -NoNewline

$sum += $number

} #foreach

#Show stats

Write-Host ("`nSum : " + $sum)

if ($n -gt 0)

{

#Display average

$avg = $sum/$n

Write-Host ("Average : " + $avg.ToString("n2"))

#Calculate STD

$sum\_d2 = 0

foreach ($number in $numbers)

{

$d2 = ($number - $avg) \* ($number - $avg)

$sum\_d2 += $d2

}

$std = [math]::Sqrt($sum\_d2/$n)

Write-Host ("STD : " + $std.ToString("n2"))

} #if

} #Try

Catch

{

Write-Host "Invalid number"

} #Catch

<# References:

https://ss64.com/ps/if.html

https://ss64.com/ps/for.html

https://ss64.com/ps/try.html

https://ss64.com/ps/get-random.html

https://ss64.com/ps/syntax-arrays.html

https://ss64.com/ps/sort-object.html

https://ss64.com/ps/syntax-math.html

https://byjus.com/standard-deviation-formula/

#>

General notes:

* You must complete and submit the declaration form. Read what you sign. A false declaration is a criminal offence and will be dealt with accordingly.
* Submit all files in a single zip file on eThuto before the due date and time. Include the declaration and class diagrams in the root folder of the submission.
* Only the last submission will be marked.