

What is purpose of following CMDLETS? Make an example script for each CMDLET.

1. Get-Help
2. Set-ExecutionPolicy
3. Get-PSDrive
4. Get-Command
5. Get-Member
6. Get-Module

Ref:

Chapter 1 and 2

Which CMDLETS are used for filtering, sorting and formatting in powershell.

Write a script which demonstrate use of

1. Filtering
2. Sorting
3. Formating

Ref:

Chapter 3 and 4

What is your understanding about

1. Profile
2. Variable
3. Array
4. Script Security
5. Sourcing
6. Variable Scope
7. Transcripts

Write a script which display brief description of each of above and store all seven definitions in txt and xml format on a local system.

Ref:

Chapter 4 and 5

Create a menu script **carinfo.ps1**. It should display five names of cars and an exit option.

1. Volvo
2. Mercedes
3. Suzuki
4. Toyota
5. Ford
6. Exit

The script should prompt user to enter a number from 1 to 6.

If the user enters 1, following message on power shell console should be displayed.

Volvo Prices

Price for 2012 model is between 100,000 DKK to 300,000 DKK

Price for 2013 model is between 200,000 DKK to 400,000 DKK

Price for 2014 model is between 200,000 DKK to 500,000 DKK

Furthermore the picture of car should also be displayed.

Do the same for other cars model. Just change the model and car prices.

If user enter 6, the script should exit

Ref:

Chapter 11 and 12

Write a script **learningfunctions.ps1**. The script should display Following menu.

1. Large Processes
2. Small Processes
3. Running Services
4. Stopped Services
5. Large Processes and Stopped Services
6. Exit

Function A:

Displays **id, name and working** set of all processes using more than **30 MB** working space

Function B

Displays **id, name and working** set of all processes using less than **30 MB** working space

Function C

Displays **total number of running services**. Furthermore it should display names of all running services in sequential order by **using for-each loop**.

Function D

Displays **total number of stopped services**.

Function E

Saves details of processes running more than 30 MB space and all stopped services into a **html** file on local hard drive.

Function F

Should exit script.

Ref:

Chapter 11 and 12

The script should contain six functions.

Function A:

Displays **id, name and working** set of all processes using more than **30 MB** working space

Function B

Displays **id, name and working** set of all processes using less than **30 MB** working space

Function C

Displays **total number** of **running services**. Furthermore it should display names of all running services in sequential order by **using for-each loop**.

Function D

Displays **total number** of **stopped services**.

Function E

Saves details of processes running more than 30 MB space and all stopped services into a **html** file on local hard drive.

Function F

Should exit script.

Ref:

Chapter 12

TASK 6

Instruction: You need three computer A, B and C to accomplish this task

1. Save Task 4 script locally computer A and Task 5 script on locally computer B.
2. From computer C create a persisted connection with computer A and computer B.
3. Execute Task 4 script and Task 5 script on computer C by using Power shell remoting respectively.

Ref:

Chatper 7



Use Task 4 script to following debugging features in code.

1. Set breakpoints on a two line numbers
2. Set a breakpoint on any variable
3. Set a breakpoint on any one command
4. Respond breakpoint with Step-into command.
5. List all break points.
6. Disable any one break point.

Ref:

Chapter13

Develop a powershell script on Windows Server 2012 R2 to

1. Configure static IP addresss
2. Install Domain Controller, DNS and DHCP service.
3. Configure DHCP scope

Place try catch finally block on script the script.

Ref:

Task 8 Document

Install MS SQL SEVER 2012 Evaluation edition on Windows Server 2012 R2.

Write a Powershell script which lists Following menu

***** SQL SERVER in POWERSHELL*****

1. SQL Server Instances
2. SQL Server configuration settings
3. Start SQL Server
4. Stop SQL Server
5. Exit

***** SQL SERVER in POWERSHELL*****

Write code behind each section of menu so that whenever executed provides a desired result.

Write a powershell script which

1. Creates a database **HR**
2. Create following table i.e. **Empolyee**

Employee Table

Id	Name	Dept	Gender	Country	Salary
1	John	IT	Male	UK	5000
2	Mary	HR	Female	India	3000
3	Todd	IT	Male	UK	3500
4	Pam	HR	Female	India	4000
5	Tatcher	Payroll	Male	USA	2000
6	Sunil	IT	Male	USA	1400
7	Hari	Payroll	Male	UK	2500
8	Sunitha	HR	Female	India	4000
9	Sneha	IT	Female	India	3000
10	Ruby	Payroll	Male	UK	4600

3. Insert 10 records into above table by using **invoke-sqlcmd** CMDLET.

BOOKS:

For Task 1 to 7:

- **Windows PowerShell 3.0 First Steps Ed Wilson , Microsoft Press**

For Task 8:

- **Windows Server 2012 Automation with PowerShell Cookbook**
 - (Chapter 2) , Packt Publishing

For Task 9 and 10:

- **SQL Server 2012 with PowerShell V3 Cookbook**
 - (Chapter 2 and 3) Packt Publishing

Supplementary Material:

Windows PowerShell Quick Reference

<http://www.microsoft.com/en-us/download/details.aspx?id=7097>

Windows PowerShell Scripting

<https://technet.microsoft.com/en-us/scriptcenter/powershell.aspx>

Microsoft Script Center

<https://technet.microsoft.com/en-us/scriptcenter/bb410849.aspx>

MCQ Test on Tasks and Book

25 Questions (Mostly Scripts)