PowerShell Assessment Scenario

|  |  |
| --- | --- |
| **NAME** |  |
| **PS NO** |  |
| **BATCH** |  |

Note:

1. There are 15 Question in this paper.
2. **Attend any 10 questions!**
3. Paste the relevant screenshots for every question once completed.

Lab time: 04:00 hrs.

1. Write a custom powershell script with the help of 5 functions as custom cmdlets and access them using individual cmdlet.

* List current execution policy
* List HDD size information (total & free space).
* List total running processes.
* Number of processors.
* BIOS Information

1. Write a powershell script that ping to multiple websites (of your choice) by the names from a text file. Display all pingable websites in GREEN colour and non-pingable in RED colour.
2. Display the following options with dynamic values (within <> brackets) whenever you open/invoke powershell session.

* Logged in username: <username>
* Internet connectivity: <Active/Inactive>
* Hostname: <hostname>
* Current IP address: <fetch IP address>

1. Display all the services within a HTML webpage with the following customization like:

* Name (starts with “S”)
* Status (equals to “running”) &
* Start Type (Automatic)

1. Get Name, display name, Status and Start Type of all the services and convert them to JSON and XML format. Write a new PowerShell script that counts number of lines in each file (JSON & XML) displays as:

* Number of lines in JSON file: <>
* Number of lines in XML file: <>

1. Write a script that counts all the duplicates of a running process (ex: SVCHOST).
2. Connect to a single remote server and install IIS on this server.
3. Write a powershell script that takes a text as input containing 5 websites names in it and ping using 1 by 1 using powershell and stores the names of non-pingable sites in another file (non-pingable.txt).
4. Write a script to create a variable containing a list of five numbers. Use a loop to calculate the square of each number and display the results.
5. Use “non-persistent remoting” approach to create following options on remote computer(s).
   * Create a directory in C:\ drive and name it “CISLTIM”.
   * Create a file in this folder with name “PowerShell.txt”
   * Write your name and PS ID in this file
6. Combine loops and conditional statements to create a script that generates numbers from 1 to 20, checks if each number is even or odd, and writes the results to a file named `EvenOddResults.txt`.
7. List top 10 processes “**Display Names**” with high CPU utilization and store these display names in a text file.
8. Write a function that to perform validation on keywords like *“LTIMindtree”, “Coimbatore”, “Tamil Nadu”* only and print/display “**Hello <>**” message when called & shows error for any other input given by user.
9. Use PowerShell to retrieve the top five processes consuming the most CPU on your system and save their names, IDs, and CPU usage to a CSV file named `TopProcesses.csv`.
10. Write a powershell program to check if $profile is present or not. If its not present then write a code to create it. After creation, write the following message “Hello <YourPSID>” & verify.