Course Outline

Windows PowerShell Scripting and Toolmaking
Course 55039: 5 days Instructor Led

About this course
In this course, you will learn to build reusable tools by using Windows PowerShell 5.0/5.1. This course focuses on the Windows PowerShell scripting language and on the concepts and techniques needed to design tools, including and output requirements and functional requirements. You will learn functions to deal with anticipated errors using standard PowerShell patterns and practices. By course completion, you will be armed with the knowledge and skills to better administer a wide variety of server and client products and technologies that offer Windows PowerShell integration, including Microsoft Exchange Server, Microsoft Windows Active Directory Domain Services, Microsoft SharePoint Server, and more.

Audience profile
Administrators that have little or no programming experience but who have a working knowledge of Windows PowerShell and who are able to use Windows PowerShell to run complex, interactive commands. Students of this course may administer a wide variety of server and client products and technologies that offer Windows PowerShell integration, including Microsoft Exchange Server, Microsoft Windows Active Directory Domain Services, Microsoft SharePoint Server, and more.

At course completion
After completing this course, students will be able to:

- Write tools, including scripting, parameterizing commands, and providing verbose output
- Debug tools and provide error handling within tools
- Combine tools into script and manifest modules
- Create custom formatting views
- Create tools that are consistent in naming and operation with native Windows PowerShell tools

Course Outline

1. Preparing for Scripting

- Securing the Scripting Environment
- Understanding Variables and Operators
- Understanding Scripting Constructs and Scope

2. Parameterizing a Command

- Designing Parameters
- Implementing Parameters

3. Creating a Script Module

- Designing Script Modules
- Implementing Script Modules

4. Handling Errors

- Designing Error Handling

- Implementing Error Handling

5. Writing Commands that Use Pipeline Input and Output

- Understanding Pipeline Parameter Binding
- Implementing Pipeline Parameter Input
- Implementing Pipeline Parameter Input

6. Creating Hierarchical Command Output

- Designing Complex Command Output
- Implementing Complex Command Output
- Using Object Hierarchies

7. Debugging Scripts

- Designing Scripts for Debugging
- Implementing Script Debugging

8. Customizing Default Formatting

- Designing Formatting
- Implementing Custom Formatting

9. Adding Advanced Parameter Attributes and Command Documentation

- Implementing Advanced Parameter Attributes
- Implementing Help Documentation

10. Creating Controller Scripts

- Designing Script Execution
- Implementing a Controller Script

11. Creating HTML-Based Reports

- Creating Basic HTML Reports
- Creating Enhanced HTML Reports

12. Creating Basic Workflows

- Understanding Workflows
- Implementing Workflows

13. Working with XML Data

- Understanding XML
- Implementing XML Manipulation

14. Using Advanced Scripting Techniques

- Using External Functionality

- Adding Graphical User Interface Elements

15. Creating Proxy Functions

- Designing Proxy Functions
- Implementing Proxy Functions

16. Building Tools in Windows PowerShell

- Designing the Tool
- Implementing the Tool
- Testing the Tool

Lab Outline

Lab 1: Parameterizing a Command

- Identify changeable values
- Declare parameters
- Use parameters in place of changeable values
- Test the script

Lab 2: Creating a Script Module

- Creating a Script Module
- Saving the script module
- Adding a module-level variable
- Controlling module member visibility
- Testing the script module

Lab 3: Handling Errors

- Using the Try…Catch Construct
- Handling Command Errors
- Handling Non-Command Errors
- Logging Errors to a File
- Displaying Warning Messages

Lab 4: Writing Commands that Use Pipeline Input and Output

- Adding Pipeline Input Capability to Parameters
- Working with Pipeline Input
- Creating Custom Output Objects
- Outputting Objects to the Pipeline

Lab 5: Creating Hierarchical Command Output

- Retrieving and Enumerating Data
- Creating Child Objects
- Creating the Parent Object
- Displaying and Object Hierarchy
- Persisting an Object Hierarchy

Lab 6: Debugging Scripts

- Using Write-Debug
- Using PSBreakpoints

Lab 7 : Customizing Default Formatting

- Adding a Custom Type Name to an Object
- Creating a DefaultDisplayPropertySet Type Extension
- Creating a Custom View
- Adding Type Extensions and Views to Modules and Creating a Module Manifest

Lab 8: Adding Advanced Parameter Attributes and Command Documentation

- Defining Aliases and Help Messages
- Defining Parameter Validation
- Adding Comment-Based Help
- Writing a Command that Uses –Confirm and -WhatIf

Lab 9: Creating Controller Scripts

- Creating a Controller Script
- Parameterizing a Controller Script
- Testing a Controller Script
- Debugging a Controller Script

Lab 10: Creating Reports by using HTML

- Creating Reports by using HTML
- Converting Objects into HTML Fragments
- Combining HTML Fragments
- Adding Basic Formatting
- Creating Enhanced HTML Fragments
- Applying Conditional Formatting

Lab 11: Creating Basic Workflows

- Importing the PSWorkflow Module
- Converting a Function to a Basic Workflow

- Parallelizing Commands

Lab 12: Working with XML Data

- Loading XML
- Manipulating XML as an Object Hierarchy
- Selecting XML Elements by using XPath
- Modifying XML
- Saving XML

Lab 13 : Creating Proxy Functions

- Generating a Proxy Function Template
- Modifying the Template
- Using the Proxy Function
- Bypassing a Proxy Function

Lab 14: Building Tools in Windows PowerShell

- Designing the Tool
- Implementing the Tool
- Testing the Tool