ADVANCED POWERSHELL 3 DAY COURSE

Course Description

This course is a continuation of MOC 10961 or AZ-040, Automating Administration with Windows PowerShell. We will be taking a deep dive into the development of PowerShell cmdlets and modules. Time will be spent on the theory of how to design your cmdlets as well as the proper structure of the programming code to facility cmdlets that work with the PowerShell piping in keeping with community standards. We will also be exploring how to place a graphical user interface on top of our PowerShell code so we can build tools that are available to everybody. This course will feature both hands on labs and challenged exercises to help you practice and the skills presented in class and develop you coding skills as we work with both Windows PowerShell and PowerShell 7 utilizing VSCode as our development environment.

Outline

Module 1 – Getting to Know Objects

This module builds the foundation of how PowerShell represents information. We will be creating our own object as the output of or code, keeping in line with PowerShell best practices.

Lessons:

- Properties
- Methods
- Events

Labs:

- Lab 1 – Discovering Objects

Module 2 – PowerShell Functions

We will discover how functions can be used to modularize your code to help making testing and development easier, but also advancing your functions to become PowerShell cmdlets that can accept information inside of the PowerShell Pipeline.

Lessons:

- Building a basic function
- Passing data
- How to use functions
- Advanced parameterization

Labs:

- Lab 2A – Using private functions
- Lab 2B – Working with Parameters
- Lab 2C – Working with switch parameters
- Lab 2D – PowerShell Pipeline By Value
- Lab 2E – PowerShell Pipeline By Property Name
- Lab 2F – Write code to Accept Input By Property Name

Module 3 – Creating Output

Now that we have the ability to create PowerShell cmdlets, we need to be able to create output that is accepted by other cmdlets. We will be creating customize objects.

Lessons:

- Why create custom objects?
- Ways to create objects
- Naming your objects
- Accepting your objects in the pipeline

Labs:

- Lab 3A – Creating custom objects
- Lab 3B – Accepting your objects in the pipeline

Module 4 – Advanced PowerShell Modules

This lesson is about polishing your PowerShell models to a professional level. We will take control of the output system to control what is displayed as well as ensure you code follows community standards.

Lesson:

- How to create a module
- Module manifest
- Formatting file
- Reviewing your code

Labs:

- Lab 4A – Complete the formatting file

Module 5 – Advanced debugging and error handling

Code never works the first time. We will focus on techniques to help you discover why your code is not working correctly and how to handle errors properly when they occur.

Lessons:

- VSCode debugging
- Try / Catch / Finally
- How PowerShell handles errors
- Handling specific errors

Labs:

- Module 5 Lab A – Custom Error Handling

Module 6 – Final coding project

In this exercise, you will create cmdlets that work together to parse data from a threat intelligence provider to recover data that you need to investigate a security incident. You will use the cumulative knowledge they you learned in the proceeding lessons to complete this assignment.

Module 7 – Graphical Interfaces

PowerShell has great potential not only for the technical users, but for the non-technical users as well. This module is about how to put a graphical interface on top of your PowerShell code to make it usable by all. We will cover the more popular graphical control elements and sharpen our skills through a series of challenges.

Lessons:

- Installing the support software
- Events
- Forms
- Label
- Textboxes
- Buttons
- Radio buttons

- Check boxes
- Data grid views
- Scopes of memory
- Form projects
- Deploying GUIs.

Prerequisites

- Completion of MOC 10961 or AZ-040
- Experience in creating PowerShell solutions.

Experience leveraging WMI/CIM using PowerShell