

STATE UNIVERSITY OF NEW YORK
COLLEGE OF TECHNOLOGY
CANTON, NEW YORK



MASTER SYLLABUS

CITA 401 – PowerShell Scripting and Automation

For available course numbers, contact the Registrar's Office at registrar@canton.edu

CIP Code: 11.0203

For assistance determining CIP Code, please refer to this webpage

<https://nces.ed.gov/ipeds/cipcode/browse.aspx?y=55>

or reach out to Sarah Todd at todds@canton.edu

Created by: Thomas Burl

Updated by:

School: Canino
Department: Decision Systems
Implementation Semester/Year: Spring 2025

A. TITLE: PowerShell Scripting and Automation

B. COURSE NUMBER: CITA 401

C. CREDIT HOURS (Hours of Lecture, Laboratory, Recitation, Tutorial, Activity):

# Credit Hours per Week	3
# Lecture Hours per Week	3
# Lab Hours per Week	
Other per Week	

D. WRITING INTENSIVE COURSE:

Yes	
No	X

E. GER CATEGORY:

Does course satisfy a GER category(ies)? If so, please select all that apply.

[1-2] Communication	
[3] Diversity: Equity, Inclusion & Social Justice	
[4] Mathematics & Quantitative Reasoning	
[5] Natural Science & Scientific Reasoning	
[6] Humanities	
[7] Social Sciences	
[8] Arts	
[9] US History & Civic Engagement	
[10] World History & Global Awareness	
[11] World Languages	

F. SEMESTER(S) OFFERED:

Fall	
Spring	X
Fall and Spring	

G. COURSE DESCRIPTION:

This course serves as an introduction to PowerShell, a robust scripting language and automation framework developed by Microsoft. Throughout the course, learners will develop the essential skills needed to effectively utilize PowerShell for system administration, task scripting, and automation. The course will cover the basic concepts and syntax of PowerShell, such as variables, data types, operators, functions, and control structures, through hands-on exercises and assignments. Students will also become proficient in navigating and interacting with the PowerShell environment, including working with the PowerShell Integrated Scripting Environment (ISE) and the command-line interface. Additionally, they will gain proficiency in manipulating various types of data. Advanced topics covered in the course include working with modules, implementing scripting best practices, handling errors, and performing remote administration. By the end of the course, students will have a strong foundation in PowerShell

scripting and automation, enabling them to create scripts that automate tasks, streamline administration processes, and boost productivity in a Windows environment.

H. PRE-REQUISITES: CITA 180
CO-REQUISITES:

I. STUDENT LEARNING OUTCOMES:

Course Student Learning Outcome [SLO]	Program Student Learning Outcome [PSLO]	GER	ISLO & Subsets
a. Define the core concepts and components of PowerShell.			5
b. Explain how to use PowerShell to automate common administrative tasks.			5
c. Analyze and troubleshoot PowerShell scripts for errors and bugs.			5
d Synthesize PowerShell commands and functions to streamline administrative tasks.			5
e. Create advanced PowerShell scripts to perform complex tasks and processes			5
f. Design and present a PowerShell project that showcases a practical application of the language.			5

KEY	<u>Institutional Student Learning Outcomes</u> <u>[ISLO 1 – 5]</u>
ISLO #	ISLO & Subsets
1	Communication Skills Oral [O], Written [W]
2	Critical Thinking <i>Critical Analysis [CA], Inquiry & Analysis [IA], Problem Solving [PS]</i>
3	Foundational Skills <i>Information Management [IM], Quantitative Lit, /Reasoning [QTR]</i>
4	Social Responsibility <i>Ethical Reasoning [ER], Global Learning [GL], Intercultural Knowledge [IK], Teamwork [T]</i>
5	Industry, Professional, Discipline Specific Knowledge and Skills

J. APPLIED LEARNING COMPONENT:

Yes	X
No	

If yes, select [X] one or more of the following categories:

Classroom / Lab	X	Community Service	
Internship		Civic Engagement	
Clinical Practicum		Creative Works/Senior Project	
Practicum		Research	
Service Learning		Entrepreneurship [program, class, project]	

K. TEXTS:

Title: "Learn PowerShell Toolmaking in a Month of Lunches"

Author: Don Jones, Jeffery Hicks

Year Published: 2017

Publisher: Manning Publications

Title: "The PowerShell Scripting and Toolmaking Book"

Author: Don Jones, Jeffery Hicks

Year Published: 2017

Publisher: Manning Publications

L. REFERENCES:

Title: "PowerShell for Sysadmins: Workflow Automation Made Easy"

Author: Adam Bertram

Year Published: 2019

Publisher: Apress

Title: "Windows PowerShell Best Practices"

Author: Ed Wilson

Year Published: 2014

Publisher: Microsoft Press

Title: "Windows PowerShell Cookbook: The Complete Guide to Scripting Microsoft's Command Shell"

Author: Lee Holmes

Year Published: 2017

Publisher: O'Reilly Media

M. EQUIPMENT:

Computer Lab

N. GRADING METHOD:

Letter Grade A-F

O. SUGGESTED MEASUREMENT CRITERIA/METHODS:

Quizzes, Exams, Programming Assignment, and Final Projects

P. DETAILED COURSE OUTLINE:

Module 1: Introduction to PowerShell

In this module, students will be introduced to PowerShell, its history, and its importance in system administration and automation. They will also learn how to install and set up PowerShell on their machines.

Learning Objectives:

- 1. Explain the purpose and benefits of using PowerShell.**
- 2. Install and configure PowerShell on a Windows machine.**
- 3. Differentiate between PowerShell versions and their features.**
- 4. Understand the basic syntax and structure of PowerShell commands.**
- 5. Execute basic PowerShell commands and view their output.**

Module 2: Navigating the PowerShell Environment

This module focuses on navigating and interacting with the PowerShell environment.

Students will learn how to use the PowerShell Integrated Scripting Environment (ISE) and the command-line interface effectively.

Learning Objectives:

- 1. Navigate through the PowerShell ISE and command-line interface.**
- 2. Customize the PowerShell environment to suit individual preferences.**
- 3. Understand and use PowerShell profiles.**
- 4. Utilize keyboard shortcuts and other time-saving techniques in PowerShell.**
- 5. Execute PowerShell commands and scripts from different locations.**

Module 3: Variables and Data Types in PowerShell

In this module, students will learn about variables and data types in PowerShell. They will explore different variable types, how to assign values to variables, and how to manipulate data using variables.

Learning Objectives:

- 1. Understand the concept of variables and their importance in PowerShell.**
- 2. Declare and assign values to variables using different data types.**
- 3. Manipulate and convert data using variables and operators.**
- 4. Use PowerShell built-in variables and environment variables.**
- 5. Apply best practices when naming variables and using data types.**

Module 4: PowerShell Operators and Expressions

This module focuses on PowerShell operators and expressions. Students will learn about arithmetic, comparison, logical, and assignment operators, and how to use them in PowerShell scripts.

Learning Objectives:

- 1. Understand the different types of operators in PowerShell.**
- 2. Use arithmetic operators to perform mathematical calculations.**
- 3. Compare values using comparison operators in PowerShell.**
- 4. Combine conditions using logical operators.**
- 5. Assign values and manipulate variables using assignment operators.**

Module 5: Conditional Statements and Loops in PowerShell

In this module, students will learn how to use conditional statements and loops in PowerShell. They will understand the syntax and usage of if-else statements, switch statements, and different types of loops.

Learning Objectives:

- 1. Understand the purpose and usage of conditional statements in PowerShell.**
- 2. Implement if-else statements to perform different actions based on conditions.**
- 3. Use switch statements to handle multiple conditions efficiently.**

4. Differentiate between different types of loops in PowerShell.
5. Write PowerShell scripts that utilize conditional statements and loops.

Module 6: PowerShell Functions and Modules

This module focuses on PowerShell functions and modules. Students will learn how to create reusable code in the form of functions, organize functions into modules, and import/export modules in PowerShell.

Learning Objectives:

1. Understand the concept and importance of functions in PowerShell.
2. Create custom functions in PowerShell to perform specific tasks.
3. Organize functions into modules for better code organization and reusability.
4. Import and use existing PowerShell modules in scripts.
5. Export and share custom PowerShell modules with others.

Module 7: Working with Files and Folders in PowerShell

In this module, students will learn how to work with files and folders using PowerShell. They will explore various file and folder manipulation techniques, including creating, deleting, copying, and renaming files and folders.

Learning Objectives:

1. Navigate through the file system using PowerShell commands.
2. Create, delete, copy, and rename files and folders using PowerShell.
3. Filter and search for specific files based on criteria.
4. Manipulate file content using PowerShell commands.
5. Understand and use PowerShell's security features for file and folder management.

Module 8: PowerShell Remoting and Scripting

This module focuses on PowerShell remoting and scripting. Students will learn how to remotely manage systems using PowerShell, execute scripts on remote machines, and automate administrative tasks.

Learning Objectives:

1. Understand the concept of PowerShell remoting and its benefits.
2. Enable and configure PowerShell remoting on local and remote machines.
3. Execute commands and scripts on remote machines using PowerShell remoting.
4. Automate administrative tasks using PowerShell remoting and scripting.
5. Troubleshoot common issues related to PowerShell remoting.

Module 9: Error Handling and Debugging in PowerShell

In this module, students will learn how to handle errors and debug PowerShell scripts. They will understand different error handling techniques and use PowerShell's debugging tools to identify and fix issues.

Learning Objectives:

1. Understand the importance of error handling in PowerShell scripts.
2. Implement error handling techniques, such as try-catch blocks.
3. Use PowerShell's built-in error handling features and cmdlet parameters.
4. Debug PowerShell scripts using breakpoints, stepping, and variable inspection.
5. Troubleshoot and fix common errors in PowerShell scripts.

Module 10: PowerShell Modules and Scripting Best Practices

This module focuses on PowerShell modules and scripting best practices. Students will learn how to create modular and well-structured scripts, follow coding standards, and ensure script security.

Learning Objectives:

1. Understand the benefits of using PowerShell modules for script organization.

- 2. Create and publish custom PowerShell modules for reuse.**
- 3. Follow scripting best practices, including code readability and documentation.**
- 4. Implement script security measures, such as script signing and execution policies.**
- 5. Collaborate and share PowerShell scripts effectively within a team.**

Q. LABORATORY OUTLINE: