

# Specialized - Perl for System Administration

---

<b>Code:</b>	TTPS2647
<b>Length:</b>	4.5 days
<b>URL:</b>	<a href="#">View Online</a>

---

This course leads the student from the basics of writing and running Perl scripts to more advanced features such as file operations, report writing, the use of regular expressions, working with binary data files, and using the extensive functionality of the standard Perl library. Once these skills are mastered, this intensive course moves into practical tools for automating Linux/Unix system administration. Students will learn how to convert existing bash scripts to Perl, and create new scripts that will safely and securely automate system administration tasks.

## Skills Gained

- Create a working script that gets input from the command line, the keyboard, or a file
- Use arrays to store and process data from files
- Create formatted reports
- Launch external programs
- Interface with bash scripts
- Understand and use regular expressions
- Use the appropriate types of variables and data structures
- Refactor duplicate code into subroutines and modules
- Know what is available in the standard library

## Who Can Benefit

This course is appropriate for anyone who wants to automate and simplify common system administration tasks with Perl.

## Prerequisites

Students should already have a working, user-level knowledge of a Unix-like operating system (Linux, Solaris, Mac OS, e.g.). While not mandatory, basic skills with at least one other programming language are desirable.

## Course Details

### Getting Started

- What is Perl?
- Structure of a Perl program
- Running a Perl script
- Checking syntax and warnings

- Execution of scripts under Unix and Windows

## **Basic Data and I/O**

- Numeric and Text literals
- Math operators and expressions
- Scalar variables
- Default values
- Writing to standard output
- Command line arguments
- Reading from the standard input

## **Logic and Loops**

- About flow control
- The if statement and Boolean values
- Using unless and elsif
- Statement modifiers
- warn() and die()
- The conditional construct
- Using while loop and its variants
- Using the for loop
- Exiting from loops

## **Lists and Arrays**

- The list data type
- Accessing array elements
- Creating arrays
- List interpolation
- Arrays and memory
- Counting elements
- Iterating through an array
- List evaluation
- Slices and ranges

## **Reading and writing text files**

- File I/O Overview
- Opening a file
- Reading text files
- Writing to a text file
- Arrays and file I/O
- Using the <> operator

## **List functions**

- Growing and shrinking arrays
- The split() function
- Splitting on whitespace
- Assigning to literal lists
- The join() function
- The sort() function
- Alternate sort keys
- Reversing an array

## **Formatting output**

- Report formatting overview
- Defining report formats
- The write() function

## **Hashes**

- Hash overview
- Creating hashes
- Hash attributes
- Testing for existence of elements
- Deleting hash elements
- Traversing a hash

## **References and data structures**

- What is a reference?
- The two ways to create references
- References to existing data
- References to anonymous data
- Dereferencing scalar, array, and hash references
- Dereferencing elements of arrays and hashes
- Multidimensional arrays and other data structures
- Sorting data structures

## **Text and Regular Expressions**

- String length
- String functions and operators
- Quoting in Perl
- Regular expressions
- Advanced pattern matching and substitution

## **Subroutines and variable scope**

- Understanding packages
- Package and Lexical variables

- Localizing builtin variables
- Declaring and calling subroutines
- Calling subroutines
- Passing parameters and returning values

## **Managing files and directories**

- File attributes
- Reading a directory
- Using File::Find
- Permissions
- Copying and moving files
- Renaming and deleting files and directories
- File test operators

## **Sys Admin Part 1**

- Environment variables
- Running external programs
- User identification
- Trapping signals
- Backtick operator
- Time of day

## **Shortcuts and defaults**

- Understanding \$\_
- shift() with no array specified
- Text file processing
- Using grep() and Using map()
- Command-line options for file processing

## **Using Modules**

- Perl modules
- Use vs. require
- A quick tour
- Bundled libraries and modules
- Downloading and installing from CPAN
- Case study: Using Getopt::Long

## **Creating Modules**

- Review of subroutines
- Understanding my () and local ()
- Packages and the symbol table
- Mechanics of module creation

- Exporting subroutines and data
- Using BEGIN and END
- Good module design

## **Classes**

- Perl's approach to Classes
- Understanding bless
- Constructors
- Methods
- Properties
- Inheritance

## **Exception handling**

- Simple exception handling
- Trapping exceptions
- Using Carp, Croak, and Cluck
- Raising exceptions

## **Sys Admin Part 2**

- Monitoring files
- Accessing syslog
- Network information
- User management

## **Network Tools**

- Copying files across the network
- Remote login
- Grabbing data from the Web

## **Serializing Data**

- Available tools
- Storable
- JSON

## **Enterprise Development**

- Coding standards
- Source code tools
- Core modules
- Saving time with module-starter
- Documenting with POD
- Benchmarking and Profiling
- Unit testing

- Distributing and installing modules

---

## Schedule (as of 3 )

Date	Location
------	----------

---

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy  
Get Your Free Copy Now

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively  
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.  
All Rights Reserved.

Generated 9