Shell Programming

Course ID #: 0070-100-ITC-W

Hours: 21

Course Content

Course Description:

Shell programs, or scripts, are the means by which a UNIX™ shell is used as a programming language. UNIX commands and shell language control constructs are entered into a file by the programmer, then the file is executed as a command and interpreted just as if the commands had been typed on the shell command line. Thus, shell scripts provide a way to automate commonly executed groups of commands - but shell scripts can do much more than this. Although many simple tasks are automated with small scripts, large scripts hundreds of lines long are very common. These larger scripts are written by system administrators, database administrators, testers, utility programmers, and others to create utilities that are largely composed of powerful UNIX commands, such as find, sed, awk, and hundreds of others. In this course, students learn to read, write, and debug Korn shell scripts. Back at work they can greatly increase productivity by automating repetitive tasks (for themselves or others), and by creating specifically tailored utilities designed to meet their precise needs. Students will read and write many shell scripts in this class, which will additionally increase their overall UNIX knowledge and skills.

Target Student:

This course is intended for UNIX users, programmers, and system administrators.

Prerequisites:

Fundamentals of UNIX

Deliver Method:

This course is delivered through a mix of instructor-led training (ILT) and hands-on labs.

Topics:

Course Introduction

- Course Objectives
- Course Overview
- Using the Workbook
- Suggested References

UNIX Processes

- What is a Process?
- Process Structure
- The ps Utility
- Options to the ps Utility
- Background Commands (&)
- Killing Background Processes
- Redirecting the Standard Error

Shell Programming

Course ID #: 0070-100-ITC-W

Hours: 21

Getting Started
- What is a Shell?
- Running Scripts
- Specifying the Script's Interpreter
- The PATH Environment Variable
- Sub-shells

Variables
- Shell Variables
- The read Command
- The export Command
- The Shell Environment
- Variable Substitution
- Command Substitution

The Login Process
- The Login Process
- The System Profile Script
- Your .profile Script
- The . Command

Conditional Statements
- The Exit Status of Commands
- Command Line Examples
- The test Command
- The if-then-else Construct
- The elif Construct
- case Statements

Loops
- The for Loop
- The while Loop
- Reading Lines From Files
- Using Arrays with Loops

Special Variables
- $$ - PID of Shell
- Command-Line Arguments
- $# - Number of Arguments
- $* - All Arguments
- The shift Command
- The set Command
- Getting Options

Quoting Mechanisms
- Single vs. Double Quotes
- What is a Here Document?
- Using a Here Document
- Here Document Quoting
- Ignoring Leading Tabs

Functions
- Shell Functions
- Passing Arguments to Functions
- Returning Values from Functions
- Function Libraries

Advanced Programming
- Shell Arithmetic
- The select Statement
- Terminal Independence in Scripts
- The eval Command

Debugging Techniques
- Using echo
- Using Standard Error
- Options for Debugging
- Script Tracing
- Conditional Debugging

Shell IPC
- Co-processes
- The print and read Commands
- Signals
- The trap Command
- Named Pipes
- The wait Command