

## HP-UX Essentials and Shell Programming Course Summary

**Length:** 5 Days

**Prerequisite:** Basic computer skills

**Recommendation Statement:**

Student should be able to use a computer monitor, keyboard, mouse. Student needs to be able to input commands and control characters from a keyboard.

**Course Description:**

For those new to HP-UX, this is the first course you will take.

The course starts at the beginner level and works into intermediate and advanced HP-UX commands and utilities. The student is introduced to the HP-UX Korn shell, the command line and command syntax. Although you will be exposed to the various windowing platforms available in HP-UX, this course teaches UNIX at “the command line level” to develop a thorough understanding of the HP-UX operating environment.

The student learns how to navigate the HP-UX file systems and how to work with files, directories and permissions. The student will learn how to use the HP-UX editors “ex” and “vi.”

Once the student is comfortable with basic HP-UX commands, they’ll move onto advanced power tools as well as how to customize the POSIX Korn shell. The student will learn how to combine shell variables, shell built-ins and UNIX power tools to build complex POSIX (Korn) shell scripts.

The classroom labs are run in a real-world HP-UX environment, are structured to allow you to learn by doing, and are developed to simulate “real world” situations. The hands-on labs progressively build your HP-UX knowledge and command skills in a clear and concise manner.

Working in a controlled UNIX classroom environment with an expert instructor, you will learn HP-UX concepts and commands. You will receive professional tips and techniques that will help you build your HP-UX confidence and skills.

**Upon completion of this course, you should be able to:**

- Log in and log out of HP-UX
- Use the Common Desktop Environment tools (CDE) and GNome window system
- Compose command-line strings to perform functions
- Navigate the directory tree
- Manipulate text files
- Create files and directories
- Backup and restore user files and directories
- Change permissions of files and directories
- Use the vi text editor
- Identify and modify initialization files
- Employ shell features to streamline command execution
- Use basic network commands
- Use commands to search directories and files
- List active user processes and selectively kill user processes
- Customize the shell environment
- Use UNIX network utilities

- Use UNIX power tools to process data and files Interpret and write complex shell scripts
- Read, write, and debug scripts
- Use local and environmental variables and shell metacharacters in scripts
- Use regular expression characters with the grep, sed, and awk utilities
- Write sed scripts to perform non-interactive editing tasks
- Write awk scripts to manipulate individual fields within a record and to write reports based upon an input file
- Use the exit status of a command to determine if the command succeeded or failed
- Access and process command-line arguments passed into a script
- Develop messages to display when a script is used incorrectly
- Use flow control constructs, such as branching and looping
- Perform string manipulation and integer arithmetic on shell variables

# HP-UX Essentials and Shell Programming

## Detailed Course Outline

### **Introduction to the UNIX Operating System**

- Describe the HP-UX Operating Environment and OS releases
- UNIX standardization - an overview of standards in the UNIX community
- Describe Unix shells (which shells are available, compare the common Unix shells, how to choose a shell)
- Describe the Unix kernel

### **Establishing Communication**

- Logging in and starting a Unix session
- Window environments available to the UNIX user
- Understanding the UNIX command syntax
- Using the UNIX "help" facility: manual pages
- Using emulation programs to access the system (ie. Putty / X-Windows / Telnet / CDE - Common Desktop Environment / Gnome Desktop )
- Setting your terminal environment (stty)
- Executing commands from the command line; UNIX Command syntax
- Using the UNIX "help" facility: manual pages

### **Introduction to the Filing System**

- Examining UNIX files and directories
- Describe the UNIX file system and directory structure
- Navigating the UNIX directory hierarchy
- Understand the difference between full and relative paths
- Understanding the current and parent directories
- Understanding the differences between a Windows/DOS based filesystem and a UNIX JFS
- Differences between Windows (DOS) files and Unix files
- Describe mount points - mounting and unmounting filesystems
- Describe hard and symbolic links
- Describe home directories
- Disk space and checking the status of disks, filesystems, and directories (ie. du and df commands)

### **File Management**

- Creating directories
- Copying files and directories
- Moving and renaming files and directories
- Removing files and directories
- Using shell metacharacters
- Locating files by name, owner, type and other parameters using the 'find' command

### **UNIX Permissions**

- Understanding permissions
- File permissions
- Directory permissions
- Change permissions with symbolic mode
- Change permissions with absolute (octal) mode
- Determine and set the umask value

### **Redirection Filters and Pipes**

- Exploring standard input, standard output, and standard error
- Using filters
- Using pipes

## Using the vi Editor

- Understand the UNIX editors
- Understand the VI modes
- Creating/viewing/manipulating/saving/deleting files using the vi editor
- Using a .exrc file

## Introduction to shell scripts

- Creating simple shell scripts
- Choosing a Shells
- UNIX Shells and Variables
- Understanding the UNIX shells
- Displaying the default variables
- Creating/Removing user variables
- stty to set terminal types (backspace, etc.)

## Introduction to the POSIX shell

- Shells: Bourne, Korn, C, bash
- Startup Scripts (.profile, .kshrc)
- Using ENV for automatic environment tailoring
- Use Bourne and Korn shell variables
- Assign values to shell variables
- Display the value of shell variables
- Make variables available to subprocesses using the export statement
- Display the value of environment variables
- Unset shell and environment variables
- Customize the user environment using the .profile file
- Perform arithmetic operations
- Create and use aliases
- Display aliases and the values assigned to them
- Display the built-in aliases
- Customize the Bourne and Korn shell environments
- Use the tilde expansion and command substitution features of the Korn shell

## Using Aliases and Functions

- Introducing the alias concept
- Using functions
- Making your aliases and functions permanent
- Removing aliases and functions

## Command Editing and Command History

- Enabling command editing in the Korn shell
- Using history to recall commands
- The command history file

## System Status and Command Information

- Displaying system information and user status
- Determining command locations

## Multitasking Capabilities

- Introduction to multitasking in the UNIX environment
- Managing jobs and background processes
- Using the process table to manage processes
- Introducing delayed and detached jobs
- Display system processes
- Use processes and process identifiers (PIDs)

- Identify parent and child processes
- Terminate processes using the kill command
- Use the pgrep and pkill commands

#### **Advanced File Management**

- Creating links
- Processing files with "awk", "tr", and "sed"
- Use awk/nawk commands from the command line
- Write awk programs to generate data reports from text files
- Write awk programs to generate numeric and text reports from text files

#### **Command line data processing**

- Using awk to display file contents and process data
- Formatting files with the tr command
- Using UNIX filters to filter, sort and restructure data
- Use awk to scan text files or standard input to display specific data, change data format, and add text to existing data

#### **Archiving User Data**

- Managing file space
- Archiving files and directories
- Compressing files
- Use the tar utility to store files
- Backup files with the tar and compress commands
- Use compression tools to save storage space (compress, gzip, etc)

#### **Printing**

- lp, lpstat, cancel

#### **Remote Connections**

- Connect remotely using telnet and ssh
- Remotely access a system using rlogin or ssh
- Transfer files with ftp and sftp (using the command line)

## Advanced Shell Topics - Day 3

### **Shell Scripts**

- Writing and Running Shell Scripts
- Command Substitution
- Shell Variables and Variable Substitution
- Shell Environment Variables
- Comments
- Getting User Input - Read
- The for Loop and if Statement
- Exit Status or Return Code
- The test Command
- The case Statement
- Simple Conditionals
- The set Command
- Arithmetic with shell variables

### **Remote Connections**

- Connect remotely using telnet and ssh
- Transfer files with ftp and sftp

### **Advanced File Management**

- Creating links
- Processing files with "awk", "tr", and "sed"
- Use awk/nawk commands from the command line
- Write awk programs to generate data reports from text files
- Write awk programs to generate numeric and text reports from text files

### **Regular Expressions and the grep Command**

- Use and describe regular expressions
- Describe the grep command
- Use the grep command to find patterns in a file
- Use the regular expression characters with the grep command
- Metacharacters
- The grep command

### **Command line database processing**

- Using awk to display file contents and process data
- Formatting files with the tr command
- Using UNIX filters to filter, sort and restructure data
- Use awk to scan text files or standard input to display specific data, change data format, and add text to existing data

### **Stream Editing with 'sed'**

- Use sed to perform noninteractive editing tasks
- Substituting text
- Deleting and printing lines
- Reading and writing files
- Multiple 'sed' editor functions
- Use regular expression characters with the sed command

## Writing and Debugging Scripts

- Creating a shell script
- Put comments in a script
- Change permissions on a script
- Execute a script
- Debug a script

## String Manipulation

- String Comparison
- String Relations
- Concatenation
- Substring Manipulations
- Wildcards and Pattern Matching

## Conditionals

- Use the exit status of a command as conditional control
- Use the "if" statement to test a condition
- Pass values using command-line arguments (positional parameters) into a script
- Create USAGE messages
- Place parameters on the command line
- Use conditional if, then, elif, else, and fi constructs
- Use exit, let, and test statements ([[ ]], " ")
- Apply the &&, ||, and ! Boolean logic operators
- Use the case statement

## Interactive Scripts

- Use the print and echo commands to display text
- Use the read command to interactively assign data to a shell variable
- Read user input into one or more variables, using one read statement
- Use special characters, with print and echo, to make the displayed text more user friendly
- Create a "here" document
- Use file descriptors to read from and write to multiple files

## Loops

- Write scripts that use for, while, and until loops
- Write a script using the select statement
- Describe when to use loops within a script
- Generate argument lists using command, variable, and file-name substitution

## Advanced Variables, Parameters, and Argument Lists

- Declare strings, integers, and array variables
- Manipulate string variables
- Change the values of the positional parameters using the set statement within a script
- Pass values using command-line arguments (positional parameters) into a script
- Declare korn shell string and array variables
- Setting and retrieving arrays
- Set default values for parameters
- Use the Korn shell built-in let, print, set, and typeset statements
- Reading user input
- Positional parameters
- Creating positional parameters
- Conditional substitution
- Working with arithmetic operators and data

## Functions

- Create user-defined functions in a shell script
- Create, invoke, and display functions from the command line
- Pass arguments into a function
- Call functions from special (function) files that are saved in one or more function directories
- Describe where functions are available for use

#### **Controlling Script Behavior**

- Manage the use of temporary files
- Write scripts using the setuid bit
- Use the crontab command

#### **Scheduling Execution**

- sleep
- date
- cron and crontab

#### **Traps**

- Describe how the trap statement works
- Include trap statements in a script
- Use the trap statement to catch signals and handle errors