Course Outline

Introduction to Perl

Course Summary

Description

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.

This is a hands-on programming class. All concepts are reinforced by informal practice during the lecture followed by graduated lab exercises which provide practice in the topics just discussed

Objectives

By the end of the course, students will be able to:

- 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
- 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
- Be able to use shortcuts and defaults, and what they replace

Topics

- Executing programs
- Variables and constants
- Conditions and iterations
- Arrays and hashes
- Working with Files
- Output formatting
- References and data structures
- Regular Expressions
- Subroutines
- Working with the Operating System
- Shortcuts and Defaults
- Using the Standard Library

Audience

This course is appropriate for anyone who wants to create applications or modules to automate and simplify common tasks with Perl.

Prerequisite

Students should already have a working, user-level knowledge of an operating system such as UNIX or Windows. While not mandatory, basic skills with at least one other programming language are desirable

Duration

Five Days

Course Outline

Introduction to Perl

Course Outline

I. *An Overview of Perl*
- A. What is Perl?
- B. Perl is compiled and interpreted
- C. Perl advantages and disadvantages
- D. Downloading and installing Perl
- E. Which version of Perl
- F. Getting help

II. *Creating and Running Perl programs*
- A. Structure of a Perl program
- B. Running a Perl script
- C. Checking syntax and warnings
- D. Execution of scripts under UNIX and Windows

III. *Basic Data and I/O*
- A. Numeric and text literals
- B. Math operators and expressions
- C. Scalar variables
- D. Default values
- E. Writing to standard output
- F. Command line arguments
- G. Reading from the standard input

IV. *Logic and Loops*
- A. About flow control
- B. The if statement and Boolean values
- C. Using unless and elsif
- D. Statement modifiers
- E. warn() and die()
- F. The conditional construct
- G. Using while loop and its variants
- H. Using the for loop
- I. Exiting from loops

V. *Lists and Arrays*
- A. The list data type
- B. Accessing array elements
- C. Creating arrays
- D. List interpolation
- E. Arrays and memory
- F. Counting elements
- G. Iterating through an array
- H. List evaluation
- I. Slices and ranges

VI. *Reading and Writing Text Files*
- A. File I/O overview
- B. Opening a file
- C. Reading text files
- D. Writing to a text file
- E. Arrays and file I/O
- F. Using the <> operator

VII. *List Functions*
- A. Growing and shrinking arrays
- B. The split() function
- C. Splitting on whitespace
- D. Assigning to literal lists
- E. The join() function
- F. The sort() function
- G. Alternate sort keys
- H. Reversing an array

VIII. *Formatting Output*
- A. Using sprintf() and printf()
- B. Report formatting overview
- C. Defining report formats
- D. The write() function
- E. Advanced filehandle magic

IX. *Hashes*
- A. Hash overview
- B. Creating hashes
- C. Hash attributes
- D. Traversing a hash
- E. Testing for existence of elements
- F. Deleting hash elements

X. *References*
- A. What is a reference?
- B. The two ways to create references
- C. References to existing data
- D. References to anonymous data
- E. Dereferencing scalar, array, and hash references
- F. Dereferencing elements of arrays and hashes
- G. Multidimensional arrays and other data structures

Course Outline

Introduction to Perl

Course Outline (cont.)

XI. *Text and Regular Expressions*
A. String length
B. The substr() function
C. The index() and rindex() functions
D. String replication
E. Pattern matching and substitution
F. Regular expressions

XII. *Raw File and Data Access*
A. Opening and closing raw (binary) files
B. Reading raw data
C. Using seek() and tell()
D. Writing raw data
E. Raw data manipulation with pack() and unpack()

XIII. *Subroutines and Variable Scope*
A. Understanding packages
B. Package and lexical variables
C. Localizing built-in variables
D. Declaring and calling subroutines
E. Calling subroutines
F. Passing parameters and returning values

XIV. *Working with the Operating System*
A. Determining current OS
B. Environment variables
C. Running external programs
D. User identification
E. Trapping signals
F. File test operators
G. Working with files
H. Time of day

XV. *Shortcuts and Defaults*
A. Understanding $_
B. shift() with no array specified
C. Text file processing
D. Using grep() and Using map()
E. Command-line options for file processing

XVI. *Data Wrangling*
A. Quoting in Perl
B. Evaluating arrays
C. Understanding qw( )
D. Getting more out of the <> operator
E. Read ranges of lines
F. Using m//g in scalar context
G. The /o modifier
H. Working with embedded newlines
I. Making REs more readable
J. Perl data conversion

XVII. *Using the Perl Library*
A. The Perl library
B. Old-style library files
C. Perl modules
D. Modules bundled with Perl
E. A selection of modules
F. Getting modules from ActiveState
G. Getting modules from CPAN
H. Using Getopt::Long

XVIII. *Some Useful Tools*
A. Sending and receiving files with Net::FTP
B. Using File::Find to search for files and directories
C. Grabbing a Web page
D. Some good places to find scripts
E. Perl man pages for more information
F. Zipping and unzipping files