



■ Corder Enterprises International ■



Building World Class MIS Teams, for you!

CP050 - Perl Programming

Course Description:

Perl has been described as C, awk, sed, and shell programming all wrapped into one language. Learn how to take advantage of Perl's power through examples and extensive hands-on exercises. This course introduces object-oriented programming in Perl.

Audience:

Programmers and system administrators.

Prerequisites:

Fundamentals of UNIX. C Programming is recommended.

Course Contents

Overview of Perl

- What is Perl?
- Running Perl Programs
- Example Programs

Perl Variables

- Three Types of Variables
- Variable Names and Syntax
- Variable Naming
- Lists
- Scalar and List Contexts
- The Repetition Operator

Arrays and Hashes

- Arrays
- Example - The @ARGV Array
- Array Functions
- Array Slices
- Hashes
- Hash Functions
- Scalar and List Contexts Revisited

CP050 - Perl Programming

I/O: Input Operations and File I/O Filehandles

- The open Function
- The Input Operator $\langle \rangle$
- Default Input Operator
- The print Function
- File Operation Functions
- Reading Directories

Operators

- Perl operators
- Operators, Functions and Precedence
- File Test Operators
- Assignment Operator Notations
- The Range Operator
- Quotation Operators
- Pattern Matching Operators

Flow Control

- Simple Statements
- Simple Statement Modifiers
- Compound Statements
- The next, last and redo Statements
- The for Loop
- The foreach Loop

Regular Expressions

- Pattern Matching Overview
- The Substitution Operator
- Regular Expressions
- Special Characters
- Quantifiers (*, +, ?, {})
- Assertions (^, \$, \b, \B)

CP050 - Perl Programming

Subroutines

- Overview of Subroutines
- Passing Arguments
- Local Variables
- Passing Names
- Returning Values

Quoting and Interpolation

- String Literals
- Interpolation
- Array Substitution
- Backslashes and Single Quotes
- Command Substitution
- Here Documents

References

- References
- Creating References
- Using References
- Passing References as Arguments to Subroutines
- Anonymous Composers
- Hard References as Hash Keys
- The Symbol Table

Complex Data Structures

- Two-dimensional Arrays in Perl
- Anonymous Arrays and Anonymous Hashes
- Arrays of Arrays
- Arrays of References
- A Hash of Arrays
- A Hash of Hashes

CP050 - Perl Programming

Packages and Modules

- Packages
- BEGIN and END Routines
- require vs. use
- Modules
- The bless Function

Object-Oriented Programming in Perl

- What is Object-Oriented?
- Why Use Object-Oriented Programming?
- Classes, Objects, and Methods in Perl
- Inheritance, the “is-a” Relationship
- Containment, the “has-a” Relationship
- Overloaded Operators
- Destructors

Advanced Regular Expressions

- Substrings
- Substrings in List Context
- RE Special Variables
- RE Options
- Multiline Res
- Substituting with an Expression

Binary Data Structures

- Variable-Length (Delimited) Fields
- Variable vs. Fixed
- Handling Binary Data
- The pack() Function
- The unpack() function
- The read () Function
- C Data Structures

CP050 - Perl Programming

Multitasking with Perl

- What are Single and Multitasking?
- UNIX Multitasking Concepts
- Process Creation with fork
- Program Loading with exec()
- File Descriptor Inheritance
- How UNIX Opens Files
- One-Way Data Flow - Pipes
- Final Result - Page Viewing

Sockets Programming in Perl

- Clients and Servers
- Ports and Services
- Berkeley Sockets
- Data Structures of the Sockets API
- Socket System Calls
- Generic Client/Server Models
- A Little Web Server

Appendix 1 - The Perl Distribution

- Where Can You Get Perl?
- How Do You Build Perl?
- What Gets Created and Installed?
- Differences Between Platforms

Appendix 2 - The Perl Debugger

- Overview of the Perl debugger
- Debugger Commands
- Non-Debugger Commands
- Listing Lines
- Single Stepping
- Setting and Clearing Breakpoints
- Modifying the Debugger
- The -w and -D Flags