COURSE NAME

PERL – BASIC

DURATION

4 Days

TRAINEE PREREQUISITES

Participants attending this training program should be competent in the following:

- Good experience in any programming language
- Fair knowledge of Windows/Linux/UNIX.

LAB SETUP (TO BE ARRANGED BY THE CLIENT)

1 PC/trainee with:

Hardware

- Core i3 processor
- 4Gb or above RAM
- Internet Connection

Software

- A computer with Perl 5.18 or higher version installed is recommended for each participant attending the training program
- SubLime Text2 IDE for the windows Env.

- This training program is largely lab-oriented, where each participant will have to try out various exercises provided by the instructor during the training sessions.
- Most of the key concepts will be explained by instructor by writing and demonstrating scripts in real-time. The participants are also required to try the same.

Projector & whiteboard

DAY WISE SYLLABUS

[Day 1]

- Introduction to scripting.

- Quick history of Perl.

- Quick primer on concepts relevant to scripting on

  UNIX/Linux platforms.

- A quick overview on Perl.

- Basic scripts on Perl.

- Comments and Script preludes.

- Invoking perl scripts.

- Perl one-liner scripts.

- Introduction to variables and data.

- Input/Output statements and flow control.

  - The print function

  - Using the <> operator

- Basic control structures in Perl

  - Statement blocks.

- The if/unless statement.

- The while/until statement.

- The for statement.

- Loop control statements (next, last, redo)

- Perl Datatypes

- Scalars, Lists, Hashes, References and Coderef.

- Basic operations on Perl data.

- Scalar data

- Numbers and Strings.

- Scalar variables, operators and functions.

- Scalar variable interpolation.

- Numeric and String functions.

- int(), oct(), hex(), rand()

- chomp(), chop(), chr(), ord(), lc(), uc(), ucfirst(), lcfirst()

- length(), reverse(), index(), sprintf(), substr()

- Basic process management.

- The system function.

- The backtick (`..`) operator.

- The eval function.

- exit, die and warn functions.

- Predefined/Special variables - Part 1.

- Coding conventions and standards.

Hands On (Exercises)

- Invoking Perl on Linux.

- Perl command-line options.

- "Hello world" program in Perl.

- Sample one-liner scripts.

- Standard I/O and the <> operator.

- Creating filters using perl scripts.

- Parsing command-line arguments.

- Program to simulate the UNIX env command.

[Day 2]

- References

  - Creating references.

  - Hard references and Symbolic references.

  - Scalar references.

- Arrays and List Data

  - Representation of Lists/Arrays in Perl.

  - List variables and operators

  - Iterating over lists using for/foreach constructs.

  - List operations and functions.

    - push(), pop(), shift(), unshift()

    - splice(), join(), split(), map(), grep(), sort(), reverse()

  - Parsing the command-line arguments using @ARGV array

  - List variable interpolation.

- Using List references.

- Multi-dimensional Lists (Array-of-Arrays).

- Common List programming pitfalls to avoid.

- Hashes

- Introduction to Hashes in Perl.

- Representation of Hashes.

- Hash variables and operators.

- Hash functions and operations.

  - keys(), values(), each(), exists(), delete()

- Using Hash references.

- Accessing the Environment variables using %ENV

- Hash of Lists, Hash of Hashes.

- Subroutines

- Defining and invoking subroutines.

- Subroutine arguments and return values.

- Call by value vs. reference.

- Passing lists and hashes to sub-routines.

- The AUTOLOAD subroutine.

- Variable scoping.

- Life-span of a variable.

- Defining variable scope using my, our and local

- Undefining variables using undef.

- Anonymous coderefs.

- Advanced operators and control structures

- Bare blocks.

- Case structures.

- Predefined/Special Perl variables - Part 2.

- Miscellaneous builtin functions

- Time-related functions

- localtime(), gmtime(), time(), times(), sleep(), alarm()

- dump(), defined(), formline(), scalar(), wantarray()

Hands On (Exercises)

- A basic password authentication program.

- A program to read the contents of a UNIX passwd

file and store the records in a Hash-of-Hash

allowing simple query operations.

- Programs/one-liners to simulate UNIX commands:

cat, tac, wc, nl, head, tail,

seq, date, sort and uniq.

- Program to count the frequency of words in a stream.

- Program to generate log of input data with time-stamp.

[Day 3]

- Patterns and Regular Expressions - Part 1

- Fundamental concepts.

- Simple uses of regexp.

- Patterns and substitutions.

- The match (m//) operator.

- The substitute (s//) operator.

- The transliterate (tr//) operator.

- Basic regexp patterns.

- Metacharacters and Metasymbols.

- Character classes.

- Quantifiers and Positions.

- Alternations.

- Non-greedy quantifiers.

- Capturing and Clustering.

- Creating manageable regexp patterns

- Commenting regexps using /x

- Using back-references.

- Managing multiple matches

- Anchors, look-ahead and look-behind

- Practical examples for commonly used regexp patterns

- File and directory access functions

- Opening and Closing files.

- open(), close() function

- File open modes (read, write, append, read-write).

- Reading and Writing data on files.

- Using the <> operator to read from a file-handle

- Using the print() function to write to a file-handle.

- Builtin filehandles - STDIN, STDOUT, STDERR

- The file test operator.

- The stat function.

- Handling errors.

- Globbing and handling directory tree.

- Using wildcards with <> operator

- Using the glob() function

- Opening, closing and reading directories.

- Using opendir(), readdir()

- Using IO::Dir

- File and directory manipulation functions.

- chmod, chown, umask, utime()

- mkdir, chdir, rmdir, rename, unlink

- link, readlink, lstat, chroot

- Object Oriented Perl

- A quick overview on OO concepts in Perl.

- Packages and Modules.

- An overview of the CPAN archive

- Installing modules via CPAN

- Including other modules into perl scripts

- use, require constructs.

- Instantiating objects from classes within imported modules/packages.

Hands On (Exercises)

- Programs/one-liners to simulate UNIX commands:

grep, tr.

- Program to substitute a pattern of text using

the I/O stream.

- Program to split a file based on BEGIN/END patterns.

- Program to extract parts of a HTML file.

- Program to validate time of day.

- Program to extract all email addresses from a file.

- Program to extract URL links from a HTML page.

- Program to extract specific HTML elements.

- Program to check rudimentary syntax of a HTML file.

- Program to translate a CSV file to HTML table data.

- Getting information on files.

- Simulation of UNIX/Linux commands:

cp, mv, rm, ln, pwd, stat, chmod,

chown, mkdir, rmdir, chdir, touch,

readlink, dirname, basename and

rudimentary ls command.

- Program to find the largest file in a directory

- Program to search and replace a pattern of text on all

files within a directory tree

- Program to search for duplicate files within a directory tree.

[Day 4]

- Database connectivity and Persistence

- Using tied hashes with DBM-style databases

- Using Storable module to persist complex data structures

- Using DBI to connect to SQL compliant RDBMS (MySQL)

- An overview on DBI architecture and DBD drivers

- Statement handles and Prepared Statements

- Fetching row from result-sets

- Perl debugging techniques

- Handling exceptions and run-time errors

- An overview on perl debugger

- Using Data::Dumper module

- CGI Scripting

- Introuduction to CGI Concepts

- Dynamic web pages using PERL CGI

- Handling GET & POST request

- Handling form data

- Invoking Perl Script through CGI

- Automation Unix/Windows

- Simple network operations using perl.

- Automating interactive command-line utilities using Expect module.

- Automating FTP operations using Net::FTP module.

- Checking remote hosts for reachability using Net::Ping module.

- Domain and hostname determination using Net::Domain module.

- Windows Administration with Perl

- Windows Administration Modules

- Managing XML data using the XML package.

Hands On (Exercises)

- Simple web application using CGI script

- Program to demonstrate Perl interaction with MySQL, performing CRUD operations

- Windows Administration with Perl

  - Program to Accessing the Event Log

  - Program to Starting and Stopping Services

  - Program to Keeping a Network's Clocks in Sync

  - Program to Dumping the Registry