Level 5 Diploma in Unix Networking (189) 149 Credits

Unit: Perl Programming **Exam Paper No.:** 5	**Guided Learning Hours:** 260 **Number of Credits:** 26
Prerequisites: Knowledge in Unix operating system commands.	**Corequisites:** A pass or higher in Certificate in Unix Networking or equivalence.
Aim: Perl is a programming language which is used for extracting information from a text file, printing out a report and converting a text file into some other form; hence learners will be introduced to this text manipulation scripting language. Perl shares the characteristics of other programming languages like C, shell scripting (sh), *AWK* and Lisp. Besides the above mentioned functions, Perl is also used for graphics programming, system administration, and network programming. Perl is widely used by UNIX and Windows programmers, system administrators, web and database programmers. It is an easy-to-learn, extremely powerful extensible language with add-on modules for almost every kind of task imaginable. Perhaps its most powerful feature is that Perl programs run on any operating system with few, if any, changes needed. This hands-on Perl programming unit provides a thorough introduction to the Perl programming language, enabling learners to develop and maintain portable scripts useful for system management, data manipulation, and Web CGI programming. Learners will learn: the Perl scripting language and learn how it compares to environments like Java and C++; how Perl lets you manipulate text and numbers in simple but powerful ways; how scripts can make decisions, perform repetitive operations; how to add arrays and lists to a scripting arsenal; debugging: the tools and techniques used to find and fix problems; files and directories, how Perl can read, write, and create files and folders; hashes and subroutines; regular expressions, one of Perl's most important capabilities for advanced processing of text; how to bring Perl capabilities to the World Wide Web; writing and using online Web forms and other interactive features; how Perl interacts with other programs.	
Required Materials: Recommended Learning Resources.	**Supplementary Materials:** Lecture notes and tutor extra reading recommendations.
Special Requirements: The course requires a combination of lectures, demonstrations, discussions, and hands-on labs.	
Intended Learning Outcomes:	**Assessment Criteria:**
1. Scripting language; step-by-step installation of Perl; what Perl is designed to do and its advantages.	1.1 Explain how Perl was developed 1.2 Explain uses of Perl 1.3 Demonstrate how to run Perl from the command line 1.4 Demonstrate how to write single Perl Programs 1.5 Analyse comments, command line environment and other features 1.5 Demonstrate how to use keyboard and system commands
2. Perl built-in data types: scalars, arrays and hashes; represent numbers (int, float, whatever) or string of text.	2.1 Define scalar data 2.2 Describe arithmetic operators 2.3 Define scalar variables 2.4 Describe operator precedence and associative 2.5 Describe identifier names and scalar variables 2.6 Describe Perl operators 2.7 Examine scalars and context 2.8 Describe special scalar variables

3. File input and output capabilities and the built-in file handles for standard input and standard output.	3.1 Define **stdout** 3.2 Define **stdin** 3.3 Explain **chop** and **chomp** operators 3.4 Describe to read from a file 3.5 Describe Perl input/output Flow Control Operations; flow control structures; functions related to flow control; library functions and how to include Perl libraries 3.6 Define statement block 3.7 Demonstrate how to use the **if...else** statement 3.8 Describe **elsif** branch 3.9 Describe the **unless** variation 3.10 Demonstrate how to use **while** and **until** statements 3.11 Demonstrate how to use **for** statement 3.12 Demonstrate how to use **for each** statement 3.13 Demonstrate conditional execution 3.14 Explain traditional branching constructs 3.15 Demonstrate how to practice using numbers, strings, and files 3.16 Explain traditional loops 3.17 Describe unusual loops 3.18 Demonstrate how to use advanced loop control
4. The declaration and access of array and list data structures in Perl; techniques available in Perl for creating stacks and queues	4.1 Define a list 4.2 Demonstrate how to use arrays and array variables 4.3 Explain array operators 4.4 Demonstrate how to use arrays and lists 4.5 Demonstrate how to traverse an array or list 4.6 Describe functions that operate on an array
5. Regular expression syntax for string of characters that define the pattern or patterns and features of Perl string manipulation facilities.	5.1 Describe single character patterns 5.2 Explain grouping patterns 5.3 Analyse anchoring patterns 5.4 Describe pattern precedence 5.5 Describe match and substitute operators 5.6 Describe standard regular expression metacharacters 5.7 Analyse Perl extended metacharacters 5.8 Evaluate the switch statement
6. Understand Perl string processing, text processing; manipulation with Regular Expressions; multidimensional associative arrays and creating multidimensional arrays in Perl.	6.1 Describe generalised quoting operators 6.2 Demonstrate how to perform string manipulations 6.3 Demonstrate how to perform list manipulations 6.4 Describe multidimensional arrays 6.5 Describe associative arrays/hashes 6.6 Demonstrate how to use hash operators 6.7 Demonstrate how to create and access a hash 8.8 Examine and evaluate traversing a hash

7. The use of subroutines in Perl and demonstrate how Perl subroutines can be called in a way that needs a return value.	7.1 Define a subroutine 7.2 Describe user-defined subroutines 7.3 Demonstrate how to pass variables to subroutines 7.4 Define a package statement 7.5 Demonstrate how to declare and call functions 7.6 Analyse function arguments and return values 7.7 Demonstrate building a library of functions
8. Programming file input and output; Perl file input and output capabilities; STDOUT, and STDERR representing standard input and standard output.	8.1 Describe print operation and processes 8.2 Describe file handles 8.3 Describe default and user defined file handles 8.4 Describe file meta-information (statistics) 8.5 Analyse functions that manipulate files 8.6 Demonstrate how to open filehandles for reading and writing
9. Printing a list of all files in a directory using the built-in Perl glob function; glob iterating through filename expansions, returning *undef* when the list is exhausted.	9.1 Demonstrate how to check file types and permissions 9.2 Demonstrate how to use directory handles 9.3 Demonstrate how to create and delete directories 9.4 Describe the glob function 9.5 Demonstrate writing template to output reports; tracking how many lines have been used in the report and automatically generate new pages as needed 9.6 Describe a report format 9.7 Demonstrate how to use file handles in reports 9.8 Demonstrate how to use format names 9.9 Demonstrate how to use placeholders
10. The DBI (Database Independent Interface) standard database interface module for Perl defining a set of methods, variables, conventions that provide a consistent database and Command Gateway Interface (CGI).	10.1 Describe **get** and **set** variables 10.2 Demonstrate how to access password by username or ID 10.3 Describe how to access group data by groupname or ID 10.4 Describe relational database jargon 10.5 Demonstrate how to use SQL insert commands 10.6 Demonstrate how to query DBMS 10.7 Describe how web works 10.8 Discuss current web technologies 10.9 Describe CGI 10.10 Define HTML forms 10.11 Describe relational databases 10.12 Define the standard Perl DBI (DataBase Interface) 10.13 Define DBD:MySQL (DataBase Driver for MySQL) 10.14 Describe the DBI Architecture 10.15 Explain how to use the handles that DBI

	provides to interact with databases 10.16 Describe the process of connecting and disconnecting from databases 10.17 Analyse error handling and describe some of the DBI's utility methods and functions 10.18 Describe how to manipulate data within a databases

Methods of Evaluation: A 2½-hour written examination paper with five essay questions, each carrying 20 marks. Candidates are required to answer all questions. Candidates also undertake project/coursework in Perl Programming with a weighting of 100%.

Recommended Learning Resources:
Perl Programming

Text Books	• Effective Perl Programming: Ways to Write Better, More Idiomatic Perl by Joseph N. Hall, Joshua A. McAdams and Brian D Foy ISBN-10: 0321496949 • Elements of Programming with Perl by Andrew Johnson ISBN-10: 1884777805 • Perl Programming by Mik Mann ASIN: B00654K7C6
Study Manuals	BCE produced study packs
CD ROM	Power-point slides
Software	Perl