LONDON CAPITAL COMPUTER COLLEGE

Diploma in Unix (189) – Perl Programming

<table>
<tr><td>**Prerequisites:** Knowledge in Unix operating system.</td><td>**Corequisites:** A pass or higher in Certificate in Unix Networking or equivalence.</td></tr>
<tr><td colspan="2">**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. 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, and 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 course provides a thorough introduction to the Perl programming language, enabling candidates to develop and maintain portable scripts useful for system management, data manipulation, and Web CGI programming. Simply put, Perl is one of a very few languages that is flexible and powerful enough to put design of the computer languages themselves into the hands of the average programmer. Candidates 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.</td></tr>
<tr><td>**Required Materials:** Recommended Learning Resources.</td><td>**Supplementary Materials:** Lecture notes and tutor extra reading recommendations.</td></tr>
<tr><td colspan="2">**Special Requirements:** The course requires a combination of lectures, demonstrations, discussions, and hands-on labs.</td></tr>
<tr><td>**Major Learning Outcomes:**</td><td>**Assessment Criteria:**</td></tr>
<tr><td>1. Define a script language, describe what Perl is designed to do and its advantages.</td><td>1.1 Discuss how Perl was developed
1.2 Outline uses of Perl
1.3 Be able to run Perl from the command line
1.4 Be able to write single Perl Programs
1.5 Analyse comments, command line environment
1.5 Be able to use keyboard and system commands</td></tr>
<tr><td>2. Describe Perl built-in data types: scalars, arrays and hashes.</td><td>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 Operators
2.7 Outline scalars and context
2.8 Explore special scalar variables</td></tr>
<tr><td>3. Outline file input and output capabilities and the built-in file handles for standard input and standard output</td><td>3.1 Define **stdout**
3.2 Define **stdin**
3.3 Explore **chop** and **chomp** operators</td></tr>
</table>

	3.4 Be able to read from a file
4. Demonstrate Perl input/output Flow Control Operations and Perl flow control structures.	4.1 Define statement block 4.2 Be able to use the **if...else** statement 4.3 Describe **elsif** branch 4.4 Describe the **unless** variation 4.5 Be able to use **while** and **until** statements 4.6 Be able to use **for** statement 4.7 Be able to use **for each** statement 4.8 Outline conditional execution 4.9 Discuss traditional branching constructs 4.10 Be able to practice using numbers, strings, and files 4.11 Discuss traditional loops 4.12 Explore unusual loops 4.13 Be able to use advanced loop control
5. Describe the declaration and access of Array and list data structures in Perl.	5.1 Define a list 5.2 Be able to use arrays and array variables 5.3 Outline array operators 5.4 Be able to use arrays and lists 5.5 Be able to traverse an array or list 5.6 Analyse functions that operate on an array
6. Regular expression syntax	6.1 Describe single character patterns 6.2 Explore grouping patterns 6.3 Analyse anchoring patterns 6.4 Describe pattern precedence 6.5 Describe match and substitute operators 6.6 Describe standard regular expression metacharacters 6.7 Analyse Perl extended metacharacters 6.8 Be able to simulate a switch statement
7. Describe Perl string processing, text processing and manipulation with Regular Expressions.	7.1 Describe generalised quoting operators 7.2 Be able to perform string manipulations 7.3 Be able to perform list manipulations
8. Define Multidimensional Associative Arrays and demonstrate how to create multidimensional arrays in Perl.	8.1 Describe multidimensional arrays 8.2 Describe associative arrays/hashes 8.3 Be able to use hash operators 8.4 Be able to create and access a hash 8.5 Explore traversing a hash
9. Describe the use of subroutines in Perl and demonstrate how Perl subroutines can be called in a way that needs a return value.	9.1 Define a subroutine 9.2 Describe user-defined subroutines 9.3 Be able to pass variables to subroutines 9.4 Define a package statement 9.5 Be able to declare and call functions 9.6 Analyse function arguments and return values 9.7 Outline building a library of functions
10. Describe programming file input and output and demonstrate Perl file input and output capabilities.	10.1 Explore print operations 10.2 Describe file handles 10.3 Describe default and user defined file handles 10.4 Describe file meta-information (statistics) 10.5 Analyse functions that manipulate files 10.6 Be able to open filehandles for reading and writing

11. Describe how print a list of all files in a directory using the built-in Perl glob function.	11.1 Be able to check file types and permissions 11.2 Be able to use directory handles 11.3 Be able to create and delete directories 11.4 Describe the glob function
12. Demonstrate how Perl uses a writing template to output reports. Describe how Perl will track how many lines have been used in the report and automatically generate new pages as needed.	12.1 Describe a report format 12.2 Be able to use file handles in reports 12.3 Be able to use format names 12.4 Be able to use placeholders
13. Describe how Perl access operating system data, uses database interface and Command Gateway Interface (CGI).	13.1 Describe **get** and **set** variables 13.2 Be able to access password by username or ID 13.3 Be able to access group data by groupname or ID 13.4 Describe relational database jargon 13.5 Be able to use SQL insert commands 13.6 Be able to query DBMS
14. Perl database interface	13.7 Describe how web works 13.8 Discuss current web technologies 13.9 Describe CGI 13.9 Define HTML forms

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