LONDON CAPITAL COMPUTER COLLEGE

Certificate in Unix (188) – Introduction to Shell Programming

<table>
<tr><td><b>Prerequisites:</b> Knowledge in Windows operating system.</td><td><b>Corequisites:</b> A pass or higher in Certificate in Networking or equivalence.</td></tr>
<tr><td colspan="2"><b>Aim:</b> The Shell Programming course provide candidates with the skills to read, write, and debug UNIX shell scripts. The course begins by describing simple scripts to automate frequently executed commands and continues by describing conditional logic, user interaction, loops, menus, traps, and functions. This course is intended for Unix users who have mastered the basics of a UNIX Operating Environment (OE) such as the SCO, Solaris and Linux and who would like to understand the various boot scripts and write scripts to automate the day-to-day Unix repetitive tasks. This course explores, in detail, the Bourne and Korn shell scripting languages. Topics include details of command execution, using variables in shell scripts, writing program code that uses flow control constructs, and signal handling. The course objectives are: Use flow control constructs, such as branching and looping; Customize system-wide shell initialization files; Use local and environmental variables and shell metacharacters in scripts; Use the exit status of a command to determine if the command succeeded or failed; Develop interactive scripts; write a script that uses functions ; write a script that uses a trap to catch a signal; Access and process command-line arguments passed into a script; Write sed scripts to perform noninteractive editing tasks; Write nawk scripts to manipulate individual fields within a record; Write nawk scripts to write reports based upon an input file; Perform string manipulation and integer arithmetic on shell variables; Write real world administration and reporting scripts; Use regular expressions with the <b>grep</b>, <b>sed</b>, and <b>nawk</b>; manipulate text files with <b>grep</b>, <b>sed</b>, and <b>nawk</b>.</td></tr>
<tr><td><b>Required Materials:</b> Recommended Learning Resources.</td><td><b>Supplementary Materials:</b> Lecture notes and tutor extra reading recommendations.</td></tr>
<tr><td colspan="2"><b>Special Requirements:</b> The course requires a combination of lectures, demonstrations, discussions, and hands-on labs.</td></tr>
<tr><td><b>Major Learning Outcomes:</b>
1. Describe the shell is an environment used to run commands, programs, and shell scripts.</td><td><b>Assessment Criteria:</b>
1.1 Explore how to run the current Shell
1.2 Outline setting up interactive environment
1.3 Explore how to edit and run scripts and functions
1.4 Outline Bourne, C and Korn shells
1.5 Explore script execution
1.6 Outline startup files</td></tr>
<tr><td>2. Describe the shell command-line interface (CLI) and the steps for setting Shell Environment Variables</td><td>2.1 Analyse command parts
2.2 Describe command types
2.3 Describe redirecting Standard Input and Output
2.4 Explore command sequences
2.5 Explore command groups</td></tr>
<tr><td>3. Demonstrate Shell variables and how they are created and maintained by the shell.</td><td>3.1 Describe variable terminology
3.2 Be able to evaluate and assigning values
3.3 Be able to quote values and references</td></tr>
<tr><td>4. Describe Shell predefined parameters and arguments.</td><td>4.1 Describe parameters
4.2 Explore processing option parameters</td></tr>
<tr><td>5. Demonstrate Special shell variables and the variables set internally by the shell and those set by the user.</td><td>5.1 Describe typed variables and scope
5.2 Be able to use arrays and compound variables</td></tr>
<tr><td>6. Demonstrate arithmetic, relational, boolean, string Operators and file test operators.</td><td>6.1 Be able to use the <b>if</b> command
6.2 Analyse the command operator test
6.3 Outline the if statement syntax</td></tr>
</table>

	6.4 Explore test conditions syntax 6.5 Be able to obtain types input from the keyboard 6.6 Describe shell variables
7. Demonstrate shell conditional operators framework used in if statements for conditional execution .	7.1 Be able to use the while and until loops 7.2 Explore the list and arithmetic for loops 7.3 Describe the **for** statement 7.4 Describe the **while** statement 7.5 Explore the **case** statement 7.6 Analyse the CASE and SELECT statements 7.7 Be able to use the case and select statements
8. Demonstrate how to use shell functions and explore the call shell functions procedure.	8.1 Define functions 8.2 Describe shell signals 8.3 Be able to produce shell scripts
9. Demonstrate how to find and locate can compare file names, or parts of file names, to shell patterns.	9.1 Define patterns 9.2 Outline command substitution 9.3 Describe parameter expansion 9.4 Outline data comparison utilities
10. Describe shell signals and processes and the purpose of signal processing in shells.	10.1 Define I/O processing 10.2 Outline the *read* and *printf* commands 10.3 Describe the *exec* command 10.4 Analyse trap processing 10.5 Analyse intercepting and generating traps
Methods of Evaluation: A 2-hour written examination paper with Section A and Section B. Section A has 40 multiple choice questions. Section B has three essay questions, each carrying 20 marks. Candidates are required to answer all questions. Candidates also undertake project/coursework in Introduction to Shell Programming with a weighting of 100%.	

Recommended Learning Resources:
Introduction to Shell Programming

Text Books	• Unix Shell Programming by Stephen G. Kochan and Patrick Wood ISBN-10: 0672324903 • Unix Shell Programming by Yashavant P. Kanetkar ISBN-10: 8170297532 • Unix Shell Programming by Lowell Jay Arthur and Ted Burns ISBN-10: 0471168947
Study Manuals	BCE produced study packs
CD ROM	Power-point slides
Software	Unix operating system