






Level 4 Certificate in Unix Networking (188) 119 Credits



Unit: Introduction to Shell Programming	Guided Learning Hours: 200
Exam Paper No.: 5	Number of Credits: 20
Prerequisites: Knowledge in Windows operating system.	Corequisites: A pass or higher in Certificate in Networking or equivalence.
<p>Aim: The Shell Programming unit provide learners with the skills to read, write, and debug UNIX shell scripts. The unit begins by evaluating simple scripts to automate frequently executed commands and continues by describing conditional logic, user interaction, loops, menus, traps, and functions. This unit is intended for Unix users who have mastered the basics of a UNIX Operating Environments such as the SCO, Solaris and Linux and who would like to implement and understand the various boot scripts and write scripts to automate the day-to-day Unix repetitive tasks. This unit 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 unit objectives are: using flow control constructs, such as branching and looping; customising system-wide shell initialisation files; using local and environmental variables and shell metacharacters in scripts; using the exit status of a command to determine if the command succeeded or failed; developing interactive scripts; writing a script that uses functions ; writing a script that uses a trap to catch a signal; accessing and process command-line arguments passed into a script; writing <i>sed</i> scripts to perform noninteractive editing tasks; writing <i>nawk</i> scripts to manipulate individual fields within a record; write <i>nawk</i> scripts to write reports based upon an input file; performing string manipulation and integer arithmetic on shell variables; writing real world administration and reporting scripts; using regular expressions with the grep, sed, and nawk; manipulate text files with grep, sed, and nawk.</p>	
Required Materials: Recommended Learning Resources.	Supplementary Materials: Lecture notes and tutor extra reading recommendations.
Special Requirements: The unit requires a combination of lectures, demonstrations, discussions, and hands-on labs.	
<p>Intended Learning Outcomes:</p> <ol style="list-style-type: none"> 1. The shell as an environment used to run commands, programs, and shell scripts. 2. The shell command-line interface (CLI) and the steps for setting Shell Environment Variables 3. Fundamental shell variables reserved for specific functions and how they are created and maintained by the shell. 4. Shell predefined parameters, syntax and programming arguments. 5. Special shell variables that provide information such as the process ID of the shell, the exit status of the last command. 	<p>Assessment Criteria:</p> <ol style="list-style-type: none"> 1.1 Describe how to run the current Shell 1.2 Outline setting up interactive environment 1.3 Demonstrate how to edit and run scripts and functions 1.4 Differentiate Bourne, C and Korn shells 1.5 Examine and outline script execution 1.6 Explain startup files 2.1 Analyse command parts 2.2 Describe command types 2.3 Describe redirecting Standard Input/ Output 2.4 Describe command sequences 2.5 Describe command groups 3.1 Describe variable terminology 3.2 Demonstrate how to evaluate and assigning values 3.3 Demonstrate how to quote values and references 4.1 Describe parameters 4.2 Explain processing option parameters 4.3 Demonstrate how to set default parameter value for a bash function 5.1 Define shell scripting 5.2 Describe special variables 5.3 Describe <i>typed</i> variables and scope

<p>6. Arithmetic, relational, boolean, string operators; file test operators and how they manipulate values of variables.</p> <p>7. Shell conditional operators framework used in IF statements for conditional execution and how add flow control statements that allow for more complex programs.</p> <p>8. How to use shell functions and their ability to create functions inside of a script to help with code reuse.</p> <p>9. How to find and locate can compare file names, or parts of file names, to shell patterns.</p> <p>10. Bourne Shell Scripting, debugging and handling shell signals processes and the purpose of signal processing in shells.</p>	<p>5.4 Demonstrate how to use arrays and compound variables</p> <p>5.5 Describe the variables set internally by the shell and those set by the user</p> <p>6.1 Demonstrate how to use the if command</p> <p>6.2 Analyse the command operator test</p> <p>6.3 Outline the IF statement syntax</p> <p>6.4 Describe test conditions syntax</p> <p>6.5 Demonstrate how to obtain types input from the keyboard</p> <p>6.6 Describe shell variables</p> <p>7.1 Demonstrate how to use the while and until loops</p> <p>7.2 Explain the list and arithmetic for loops</p> <p>7.3 Describe the for statement</p> <p>7.4 Describe the while statement</p> <p>7.5 Describe the case statement</p> <p>7.6 Analyse the CASE and SELECT statements</p> <p>7.7 Demonstrate how to use the case and select statements</p> <p>8.1 Define functions</p> <p>8.2 Describe shell signals</p> <p>8.3 Describe the process of producing shell scripts</p> <p>8.4 Explain the call shell functions procedure</p> <p>8.5 Demonstrate how to create shell script functions using Bash</p> <p>9.1 Define patterns</p> <p>9.2 Outline command substitution implementation</p> <p>9.3 Describe parameter expansion</p> <p>9.4 Describe data comparison utilities</p> <p>10.1 Define I/O processing</p> <p>10.2 Outline the read and printf commands</p> <p>10.3 Describe the exec command</p> <p>10.4 Analyse trap processing</p> <p>10.5 Analyse intercepting and generating traps</p>
<p>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%.</p>	

Recommended Learning Resources: Introduction to Shell Programming

<p>Text Books</p>	<ul style="list-style-type: none"> • 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
<p>Study Manuals</p> 	<p>BCE produced study packs</p>
<p>CD ROM</p> 	<p>Power-point slides</p>
<p>Software</p> 	<p>Unix operating system</p>