

STATUS:   A  

COMMON COURSE NUMBER:   CTS 1344C  

COURSE TITLE:   Sun: Advanced Shell Scripting  

CREDIT HOURS:           4          

**CONTACT HOURS BREAKDOWN:**

Lecture/Discussion	<u>          56          </u>
Lab	<u>          8          </u>
Other	<u>                          </u>
Contact Hours/Week	<u>          4          </u>

**CATALOG COURSE DESCRIPTION:**

This course provides students 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 system administrators who have mastered the basic Solaris[tm] Operating Environment (OE) and who would like to read and understand the various boot scripts and write their own scripts to automate their day-to-day tasks. This course explores, in detail, the Bourne and Korn shell scripting languages.

Prerequisite: COP1343C (with a grade of C or higher)

Corequisite:

**UNIT TITLES:**

Unit 1: UNIX Shells and Shell Scripts  
Unit 2: Writing and Debugging Scripts  
Unit 3: The Shell Environment  
Unit 4: Regular Expressions and the grep Command  
Unit 5: The sed Editor  
Unit 6: The nawk Programming Language  
Unit 7: Conditionals  
Unit 8: Interactive Scripts  
Unit 9: Loops  
Unit 10: Advanced Variables, Parameters, and Argument Lists  
Unit 11: Functions  
Unit 12: Traps

## I. Course Overview:

Upon successful completion of this course, the students should be able to read, write, and debug complex UNIX shell scripts.

## II. Units:

### Unit 1. 1. UNIX Shells and Shell Scripts

#### General Outcome:

1.0 The students should be able to describe the UNIX shell and write a shell script.

#### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

1.1 Describe the role of shells in the UNIX environment

1.2 Describe the standard shells

1.3 Define the components of a shell script

1.4 Write a simple shell script

## Unit 2. 2. Writing and Debugging Scripts

### General Outcome:

2.0 The students should be able to create, execute, and debug Unix shell scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

- 2.1 Start a script with #!
- 2.2 Put comments in a script
- 2.3 Change permissions on a script
- 2.4 Execute a script
- 2.5 Debug a script

## Unit 3. 3. The Shell Environment

### General Outcome:

3.0 The students should be able to manipulate the shell environment.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

3.1 Use Bourne and Korn shell variables

3.2 Assign values to shell variables

3.3 Display the value of shell variables

3.4 Make variables available to subprocesses using the export statement

3.5 Display the value of environment variables

3.6 Unset shell and environment variables

3.7 Customize the user environment using the .profile file

3.8 Perform arithmetic operations

3.9 Create and use aliases

3.10 Display aliases and the values assigned to them

3.11 Define the built-in aliases

3.12 Customize the Bourne and Korn shell environments

3.13 Use the tilde expansion and command substitution features of the Korn shell

## Unit 4. 4. Regular Expressions and the grep Command

### General Outcome:

4.0 The students should be able to create regular expressions, including expressions that use the grep command.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

4.1 Use and describe regular expressions

4.2 Describe the grep command

4.3 Use the grep command to find patterns in a file

4.4 Use the regular expression characters with the grep command

**Unit 5. 5. The sed Editor**

General Outcome:

5.0 The students should be able to use the sed editor.

Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

5.1 Use the sed editor to perform noninteractive editing tasks

5.2 Use regular expression characters with the sed command

## Unit 6. 6. The nawk Programming Language

### General Outcome:

6.0 The students should be able to write programs using nawk.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

6.1 Use nawk commands from the command line

6.2 Write simple nawk programs to generate data reports from text files

6.3 Write simple nawk programs to generate numeric and text reports from text files

## Unit 7. 7. Conditionals

### General Outcome:

7.0 The students should be able to use conditional shell statements.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

7.1 Use the exit status of a command as conditional control

7.2 Use the "if" statement to test a condition

7.3 Pass values using command-line arguments (positional parameters) into a script

7.4 Create USAGE messages

7.5 Place parameters on the command line

7.6 Use conditional if, then, elif, else, and fi constructs

7.7 Use exit, let, and test statements ([[ ]], " ")

7.8 Apply the &&, ||, and ! Boolean logic operators

7.9 Use the case statement

## Unit 8. 8. Interactive Scripts

### General Outcome:

8.0 The students should be able to create interactive scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

8.1 Use the print and echo commands to display text

8.2 Use the read command to interactively assign data to a shell variable

8.3 Read user input into one or more variables, using one read statement

8.4 Use special characters, with print and echo, to make the displayed text more user friendly

8.5 Create a "here" document

8.6 Use file descriptors to read from and write to multiple files

## Unit 9. 9. Loops

### General Outcome:

9.0 The students should be able to use loop control structures in their shell scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

9.1 Write scripts that use for, while, and until loops

9.2 Write a script using the select statement

9.3 Describe when to use loops within a script

9.4 Generate argument lists using command, variable, and file-name substitution

## Unit 10. 10. Advanced Variables, Parameters, and Argument Lists

### General Outcome:

10.0 The students should be able to use simple data structures and parameters in their shell scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

10.1 Declare strings, integers, and array variables

10.2 Manipulate string variables

10.3 Change the values of the positional parameters using the set statement within a script

10.4 Use Korn shell arrays

10.5 Set default values for parameters

10.6 Use the Korn shell built-in let, print, set, and typeset statements

## Unit 11. 11. Functions

### General Outcome:

11.0 The students should be able to use sub-routines in their shell scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

11.1 Create user-defined functions in a shell script

11.2 Create, invoke, and display functions from the command line

11.3 Pass arguments into a function

11.4 Call functions from special (function) files that are saved in one or more function directories

11.5 Describe where functions are available for use

## Unit 12. 12. Traps

### General Outcome:

12.0 The students should be able to use trap statements in their shell scripts.

### Specific Learning Outcomes:

Upon Successful completion of this unit, the students should be able to:

12.1 Describe how the trap statement works

12.2 Include trap statements in a script

12.3 Use the trap statement to catch signals and handle errors