UNIX PROGRAMMING			
(Effective from the academic year 2018 -2019)			
SEMESTER – V			
Course Code	**18CS56**	**CIE Marks**	40
Number of Contact Hours/Week	3:0:0	**SEE Marks**	60
Total Number of Contact Hours	40	**Exam Hours**	03
CREDITS – 3			
Course Learning Objectives: This course (18CS56) will enable students to			

- Interpret the features of UNIX and basic commands.
- Demonstrate different UNIX files and permissions
- Implement shell programs.
- Explain UNIX process, IPC and signals.

Module 1	Contact Hours
Introduction: Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. General features of Unix commands/ command structure. Command arguments and options. Basic Unix commands such as echo, printf, ls, who, date,passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the super user: su command. **Unix files:** Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands. **RBT: L1, L2**	08
Module 2	
File attributes and permissions: The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions. **The shells interpretive cycle:** Wild cards. Removing the special meanings of wild cards. Three standard files and redirection. **Connecting commands:** Pipe. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions. **Shell programming:** Ordinary and environment variables. The .profile. Read and readonly commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples. **RBT: L1, L2**	08
Module 3	
UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs. **UNIX Processes and Process Control:** **The Environment of a UNIX Process:** Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions,	08

getrlimit, setrlimit Functions, UNIX Kernel Support for Processes. **Process Control:** Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions **RBT: L1, L2, L3**	
Module 4	
Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. **Overview of IPC Methods**, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores. **Shared Memory**, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions. **RBT: L1, L2, L3**	08
Module 5	
Signals and Daemon Processes: Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model. **RBT: L1, L2, L3**	08

Course Outcomes: The student will be able to :

- Explain Unix Architecture, File system and use of Basic Commands
- Illustrate Shell Programming and to write Shell Scripts
- Categorize, compare and make use of Unix System Calls
- Build an application/service over a Unix system.

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill (Chapter 1,2 ,3,4,5,6,8,13,14)
2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005 (Chapter 3,7,8,10,13,15)
3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999. (Chapter 7,8,9,10)

Reference Books:

1. M.G. Venkatesh Murthy: UNIX & Shell Programming**,** Pearson Education.
2. Richard Blum , Christine Bresnahan : Linux Command Line and Shell Scripting Bible, 2ndEdition, Wiley,2014.

Faculty can utilize open source tools to make teaching and learning more interactive.