COURSE PLAN

Rev. No 9 Dated: September 2024

Name of School

Cluster	Systems
Program	B.Tech.
Course	Linux Lab
Course Code	CSEG1126
No. of credits	2
Semester	I
Session	Aug 2025 – Dec 2025
Academic Year	2025-26

COURSE PLAN

Prerequisite		
Credit	**2**	
Lecture	**Tutorial**	**Practical**
0	**0**	**4**

A. The expected Program Outcome are:

PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental

	contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

B. Expected Program specific Outcome are:

PSO1	Perform system and application programming using computer system concepts, the principles of data structures, algorithm development, problem-solving, and optimization techniques.
PSO2	Apply software development and project management methodologies, integrating principles from both front-end and back-end development, and effectively utilize contemporary tools and technologies.
PSO3	Exhibit a commitment to ethical practices, societal responsibilities, and continuous learning, contributing to the advancement of technology and addressing challenges in diverse computing domains.

C. **The expected Course Outcomes are:**

CO 1	Understand the functionality and purpose of different file operations, user management commands, and system information commands in Linux.
CO 2	Apply the appropriate commands to navigate the file system,

	manage files/directories, view/edit files, manage users, and gather system information.
CO 3	Analyse and identify potential issues or improvements in shell scripts by examining their logic, structure, and performance.

D. CO-PO Relationship Matrix

Indicate the relationships by1- Slight (low) 2- Moderate (Medium) 3-Substantial (high)

Program Outcomes / Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
CO 2	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
Average	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

E. Course Outcomes assessment plan:

Components / Course Outcomes	Quiz	Lab Performance	Lab Report
CO 1	☑	☑	☑
CO 2	☑	☑	☑
CO3	**£**	☑	☑
CO4	**£**	☑	☑
CO5	**£**	☑	☑

F. Course Syllabus Template

List of Experiments

Experiment 1: Install virtual player and Linux

Theory: History of UNIX, The UNIX philosophy, GUI, Overview of the Linux Operating System, Unix commands, Introduction to VI editor

Lab:

Step 1: Install VMware Workstation 15 Player or VirtualBox by Oracle.

i. VMware Workstation 15 Player is a virtualization software that allows you to run multiple operating systems on a single physical machine. You can download it from the VMware website and follow the installation instructions.
ii. VirtualBox is another popular virtualization software that provides similar functionality. You can download it from the Oracle website and install it on your computer.

Step 2: Download an installation .iso for a Linux distribution like Ubuntu, Fedora, or any other of your choice.

i. An .iso file is a disk image of the Linux distribution you want to install. It contains all the necessary files for the operating system.
ii. You can download the .iso file from the official website of the Linux distribution you prefer. For example, if you want to use Ubuntu, you can visit the Ubuntu website and download the .iso file for the desired version.

Step 3: Install the .iso from the virtual VBox or VMware Workstation.

i. Launch VMware Workstation or VirtualBox on your computer.
ii. Create a new virtual machine by clicking on the "New" or "Create" button, depending on the software you are using.
iii. Follow the on-screen instructions to set up the virtual machine. This includes specifying the name, location, and specifications for the virtual machine, such as the amount of RAM and storage allocated.

iv. During the setup process, you will be prompted to select the installation .iso file. Browse and select the .iso file you downloaded in Step 2.

v. Complete the virtual machine setup by following the remaining on-screen instructions. This may involve selecting the operating system type, configuring network settings, and specifying storage options.

vi. Once the setup is complete, start the virtual machine. It will boot from the .iso file and begin the Linux installation process.

vii. Follow the installation prompts and provide any required information, such as username, password, and partitioning options.

viii. Once the installation is complete, you will have a fully functional Linux virtual machine running within VMware Workstation or VirtualBox.

Experiment 2: Practice some basic commands on Linux

Theory: Overview of file systems in Linux, File permissions and ownership, modifying file permissions and ownership, Creating and editing files using text editors

Lab:

1. Basic Navigation Commands: Teach essential commands for navigating the file system, such as:
   ls (list): Display the contents of a directory.
   cd (change directory): Move between directories.
   pwd (print working directory): Show the current directory. mkdir (make directory): Create new directories.
   rmdir (remove directory): Remove empty directories.
2. File Operations: Introduce fundamental file operations using the following commands:
   touch: Create new files.
   cp (copy): Copy files and directories.
   mv (move): Move or rename files and directories. rm (remove): Delete files and directories.
3. File Viewing and Editing: Introduce commands for viewing and editing files, such as: cat: Display file contents.

less or more: View files with pagination.

head and tail: Show the beginning or end of a file. nano or vim: Basic text editors.

4. User Management: Discuss commands for user management tasks, including:

   whoami: Display the current user.

   who: Show users currently logged in.

   passwd: Change the password for the current user.

   sudo (superuser do): Execute commands with administrative privileges.

5. System Information: Introduce commands for gathering system information, such as:

   uname: Display system information.

   df (disk free): Show disk space usage.

   top or htop: Monitor system processes.

   history: View command history.

Experiment 3: Files and Directories commands

Theory: File Manipulation Commands, File Compression and Archiving, File Searching, File System Navigation and Management, File Transfer

Lab:

1. Working with Files:

   touch: Create an empty file or update the access/modify timestamps of an existing file.

   cp: Copy files and directories.

   mv: Move or rename files and directories. rm: Remove files and directories.

   cat: Concatenate and display the contents of a file. less: Display the contents of a file one page at a time. head: Display the first few lines of a file.

   tail: Display the last few lines of a file.

2. File Permissions and Ownership:

   Explain the ls -l command to display detailed file information, including permissions and ownership.

   Discuss the three sets of permissions: owner, group, and others. Explain the chmod command to modify file permissions.

Discuss the chown and chgrp commands to change file ownership and group.

3. Advanced File and Directory Operations:
   find: Search for files and directories based on various criteria. grep: Search for specific patterns within files.
   tar: Archive files and directories into a single file. gzip/gunzip: Compress and decompress files.
   ln: Create hard and symbolic links.

Experiment 4: Shell Programming

Theory: Introduction to BASH shell scripting, Basics of Shell Scripting, Types of Shell, Shell variable, Shell Keywords, Basic Operator, Positional Parameters

Lab:

1. Write a simple shell script that prints "Hello, World!" when executed.
2. Create a script that prompts the user to enter their name and then displays a personalized greeting.
3. Write a script that takes two numbers as input and performs various arithmetic operations like addition, subtraction, multiplication, and division.
4. Create a script that asks the user to enter their age and displays a message based on whether they are eligible to vote or not.

Experiment 5: Shell Programming

Theory: Command Line Argument, Array, Conditional Statements, Decision Making,

Lab:

1. Write a script that takes a number as input and checks whether it is a prime number or not.
2. Write a script that calculates the sum of the digits of a given number.
3. Create a script that checks whether a given number is an Armstrong number or not.

Experiment 6: Shell Programming

Theory: Shell Loops, Loop control, IO Redirections, Shell Function, Regular Expressions, Script Debugging and Troubleshooting

Lab:

1. Write a script that checks whether a given number is a palindrome or not. A palindrome number reads the same backward as forward.
2. Write a script that calculates the greatest common divisor (GCD) and the least common multiple (LCM) of two given numbers.
3. Create a script that takes multiple numbers as input and sorts them in ascending or descending order.

Experiment 7: Shell Programming

Theory: Introduction to Processes, Process states, and process hierarchy, Process Management Commands: Viewing and monitoring running processes, Terminating or killing processes, Process Prioritization and Scheduling,

Lab:

1. Write a script that takes a filename as input and checks if it exists. If the file exists, display its content; otherwise, prompt the user to create the file.
2. Create a script that prints the numbers from 1 to 10 using a loop.
3. Write a script that takes a filename as a command line argument and counts the number of lines, words, and characters in that file.
4. Create a script that defines a function to calculate the factorial of a given number and call that function with different inputs.

Experiment 8: Shell Programming

Theory: Process Control and Signals, Process Monitoring and Resource Usage, Process Communication, Process Synchronization, Background Processes and Job Control, System Monitoring and Logging

Lab:

1. Write a script that checks the file permissions of a given file and displays whether it is readable, writable, or executable by the current user.
2. Create a script that prompts the user to enter a string and then performs operations like string length, string concatenation, and string comparison.
3. Write a script that searches for a specific pattern in a given file and displays the matching lines.
4. Create a script that displays various system information like the current date and time, logged-in users, system uptime, etc.

Experiment 9: Shell Programming

Theory: System Performance Monitoring, System Security and User Management.

Lab:

1. Write a script that renames all files in a directory by adding a prefix or suffix to the filenames.
2. Create a script that searches for files in a specified directory and its subdirectories, based on certain criteria like file extension or file size.
3. Write a script that generates the Fibonacci series up to a given number, using loops or recursive functions.

Experiment 10: Shell Programming

Theory: Writing modular and reusable code, Script optimization

Lab:

1. Write a script that takes a string as input and calculates its length.
2. Create a script that takes a string as input and prints its reverse.
3. Write a script that prompts the user to enter two strings and concatenate them together.

Experiment 11: Shell Programming

Theory: Interacting with Users: Interactive shell scripts, Parsing and Processing Data Formats, Interacting with Databases

Lab:

1. Write a script that takes a sentence as input and splits it into individual words.
2. Create a script that checks whether a given string is a palindrome or not.

Experiment 12: Building a Rule-Based Expert System using Shell Scripting

Theory: Process Automation and Job Scheduling: Automating repetitive tasks using shell scripts, System Administration Scripts, Managing services and daemons.

Lab:

Objective: The objective of this lab exercise is to build a simple rule-based expert system using shell scripting. The expert system will provide recommendations based on a set of predefined rules.

Instructions:

1. Create a shell script named "expert_system.sh".
2. Implement a set of rules using conditional statements (if-elif-else) within the script. Each rule should check for specific conditions and provide a corresponding recommendation.

   Example rules:

   - If the user is experiencing fever, recommend taking a fever reducer medication.
   - If the user has a sore throat, recommend gargling with warm saltwater.
   - If the user has a cough and congestion, recommend drinking warm fluids and taking cough syrup.
   - Feel free to add more rules based on your desired expert system topic.
3. Prompt the user to input their symptoms.
4. Based on the user's input, evaluate the rules one by one and display the appropriate recommendation(s) for the symptoms identified.

5. If none of the rules match the user's symptoms, provide a general recommendation or message.
6. Test the expert system by running the script and providing different sets of symptoms to observe the recommendations.
7. Modify the rules or add new rules as needed to refine the expert system's behavior.
8. Document the logic and rules implemented in the script, along with any modifications or additions made.
9. Write a summary report discussing the challenges faced, observations made, and improvements that can be made to enhance the expert system's

G. Target

Target	
Level-1	
Level-2	
Level-3	

H. Method of Evaluation*

Components	Quiz	Performance & Viva	Lab Report	Total
Weightage (%)	30	50	20	100

*It can be revised as per the assessment scheme of the respective School/Course

I. Passing Criteria

Scale	PG	UG
Out of 10point scale	SGPA – "6.00" in each semester CGPA – "6.00" Min. Individual Course Grade – "C" Course Grade Point – "4.0"	SGPA – "5.0" in each semester CGPA – "5.0" Min. Individual Course Grade – "C" Course Grade Point – "4.0"

*for PG, passing marks are 40/100 in a paper (Composite)

*for UG, passing marks are 35/100 in a paper (Composite)

J. References:

Text Books	1. Dayanand Ambawade, and Deven Shah, "Linux Labs and Open Sou Technologies", Dreamtech Press, 2014. Paul W Browning, "101 Labs - Linux LPIC1: Includes Linux Essentials", Re Press Ltd, 2019.
Web resources	
Journals	
Reference books	
MOOCs, Online courses	

SUGGESTIONS FOR FACULTY

- Faculty should keep track of the students with low attendance and counsel them regularly.
- The course coordinator will arrange to communicate the short attendance (as per UPES policy) cases to the students and their parents monthly.
- Topics covered in each class should be recorded in the table of RECORD OF CLASS TEACHING (Suggested Format).
- Internal assessment marks should be communicated to the students twice in a semester.
- The file will be audited by respective IQAC members for theory as well as for lab as per schedule.
- The faculty is required to maintain these files for a period of at least three years.
- This register should be handed over to the head of department, whenever the faculty member goes on long leave or leaves the Colleges/University.
- For labs, continuous evaluation format (break-up given in the guidelines for result preparation in the same file) should be followed.
- The department should monitor the actual execution of the components of continuous lab evaluation regularly.
- Instructor should maintain record of experiments conducted by the students in the lab weekly.
- Instructor should promote students for self-study and to make concept diary, due weightage in the internal should be given under faculty assessment for the same.
- Course outcome assessment: To assess the fulfilment of course outcomes two different approaches have been decided. Degree of fulfillment of course outcomes will be assessed in different ways through direct assessment and indirect assessment. In Direct Assessment, it is measured through quizzes, tests, assignment, Mid-term and/or End-term examinations. It is suggested that each examination is designed in such a way that it can address one or two outcomes (depending upon the course completion). Indirect assessment is done through the student survey which needs to be designed by the faculty (sample format is given below) and it shall be conducted towards the end of course completion. The evaluation of the achievement of the Course Outcomes shall be done

by analyzing the inputs received through Direct and Indirect Assessments and then corrective actions suggested for further improvement.

- At the completion of the course, course attainment and other documents should be shared with the program coordinator for computation of Program attainment.
- At the completion of the course Faculty members are suggested to share the innovative teaching techniques along with the course plan (format provided by IQAC).
- Faculties are encouraged to share the master/expert classes evidence (as per the event report format)
- Faculties are also encouraged to include MOOCs,,SWAYAM any other online content and share the evidence of MOOCs courses /online courses referred (as per the event report format).
- Faculties are encouraged to share the evidence related to interventions or initiatives focusing the unique/slow and Fast Leaners along with Course Completion files.

INDIRECT ASSESSMENT

Sample format for Indirect Assessment of Course outcomes:

NAME:
ENROLLMENT NO:
SAP ID:
COURSE:
PROGRAM:

Please rate the following aspects of course outcomes of ----------.

Use the scale 1-3*

Course Outcomes	Statement	1	2	3
CO1				
CO2				
CO3				
CO4				
CO5				
CO6				

CO7				

1.

* 1 WEAK 2 MODERAT 3 STRONG