

Problem Solving Using C			
[As per Choice Based Credit System (CBCS) scheme]			
SEMESTER – I			
Subject Code	16MCA11	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Understand basics of C programming language</p> <p>CO2: Acquire knowledge of, Various types of control statements and structures</p> <p>CO3: Analyze the performance of, arrays, functions, pointers, structures, unions, files and preprocessor directives</p> <p>CO4: Implement all the applications of C in a high-level language</p> <p>CO5: Design and apply appropriate C language constraints for solving computing problems.</p>			
Modules			Teaching Hours
Module 1			10 Hours
Algorithms, Flow Charts, C structure, Variables, Data types, Constants, Declarations, Operators, Precedence, Associativity, Order of evaluation, Type conversion, Storage classes, Programming Examples, Input and output statements – scanf, getchar, gets, printf, putchar, puts			
Module 2			10 Hours
Control Statements – if, else-if, switch, Control Structures – while, for, do-while, break and continue, goto, Programming Examples			
Module 3			10 Hours
Arrays – Single dimension, Two dimensional, Multi dimensional Arrays, Strings, Programming Examples, Functions, Categories of functions			
Module 4			10 Hours
Pointers, Pointer arithmetic, Call by value, Pointer Expression, Pointer as function arguments, recursion, Passing arrays to functions, passing strings to functions, Call by reference, Functions returning pointers, Pointers to functions, Programming, Examples, Structures and Unions – defining, declaring, initialization, accessing, comparing, operations on individual members; array of structures, structures within structures, structures and functions, pointers and structures, bit fields, Programming Examples			
Module 5			10 Hours
Files – defining, opening, closing, input and output operations, error handling, random access; Command line arguments; Dynamic Memory Allocation – definition, malloc, calloc, realloc, free, dynamic arrays; Preprocessor – definition, macro substitution, file inclusion, compiler control directives, Programming Examples			
<p>Question paper pattern:</p> <ul style="list-style-type: none"> · The question paper will have ten questions. · Each full question consists of 20 marks. · There will be 2 questions from each module. · Each question will have questions covering all the topics under a module. <p>The students will have to answer 5 full questions, selecting one full question from each module</p>			

Text Books

1. Let us C, Yashwant Kanetkar, BPB Publications
2. Programming with C, Balaguruswamy

The C Programming Language, Brian W Kernighan, Dennis M Richie, PHI, 2nd Edition

Reference Books:

1. Programming with C, Byron Gottfried, Tata McGraw-Hill edition
2. Simplifying C, Harshal Arolkar, Sonal Jain, Wiley Publications
3. Head First C, David Griffiths, & Dawn Griffiths, O'Riley.

C Programming, Dr. Vishal M. Lichade, Dreamtech press

UNIX Programming [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA12	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Understand and experience the UNIX environment, File system and hierarchy.</p> <p>CO2: Demonstrate commands to extract, interpret data for further processing.</p> <p>CO3: Apply commands to perform different tasks using AWK filtering for various applications</p> <p>CO4: Analyze the usage of different shell commands, variables and analyzing the working of Source Code Control System Tool-GIT with Example</p> <p>CO5: Evaluate different commands with sample shell scripts</p>			
Modules			Teaching Hours
Module 1			10 Hours
<p>Introduction of UNIX and Shell: Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tput, bc, script, spell and ispell, Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators && and , exit, if, and case conditions, expr, sleep and wait, while, until, for, \$, @, redirection. The here document, set, trap, Sample Validation and Data Entry Scripts.</p>			
Module 2			10 Hours
<p>UNIX File System: The file, what's in a filename? The parent-child relationship, pwd, the Home directory, absolute pathnames, using absolute pathnames for a command, cd, mkdir, rmdir, Relative pathnames, The UNIX file system. Basic File Attributes: ls -l, the -d option, File Permissions, chmod, Security and File Permission, users and groups, security level, changing permission, user masks, changing ownership and group, File Attributes, More file attributes: hard link, symbolic link, umask, find.</p>			
Module 3			10 Hours
<p>Filters and Awk Pr, head, tail, cut, paste, sort, uniq, tr commands, Filters using Regular Expression : grep & sed grep, Regular Expression, egrep, fgrep, sed instruction, Line Addressing, Inserting and Changing Text, Context addressing, writing selected lines to a file, the -f option, Substitution, Properties of Regular Expressions Context addressing, writing selected lines to a file, the -f option, Substitution, Properties of Regular Expressions. Awk-Advanced Filters: Simple awk Filtering, Splitting a Line into Fields, printf, the Logical and Relational Operators, Number Processing, Variables, The -f option, BEGIN and END positional Parameters, get line, Built-in variables, Arrays, Functions, Interface with the Shell, Control Flow.</p>			

Module 4	10 Hours
<p>Advanced Shell Programming The sh command, export, cd, the Command, expr, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement and Examples</p> <p>Source Code Control Systems- GIT Version Control Systems, Distributed version Control Systems, Advantages of GIT, DVCS Terminologies, Life Cycle, Create Operation, Create New User, Create a Bare Repository</p>	
Module 5	10 Hours
<p>Process and System Administration Process basics, PS, internal and external commands, running jobs in background, nice, at and batch, cron, time commands, Essential System Administration root, administrator's privileges, startup & shutdown, managing disk space, cpio, tar, Customizing the Environment : System Variables, profile, sty, PWD, Aliases, Command History, On-line Command Editing. Advanced System Administration: Case Study: emacs editor and any one distribution of Linux</p>	
<p>Question paper pattern:</p> <ul style="list-style-type: none"> · The question paper will have ten questions. · Each full question consists of 20 marks. · There will be 2 questions from each module. · Each question will have questions covering all the topics under a module. <p>The students will have to answer 5 full questions, selecting one full question from each module.</p>	
<p>Text Books:</p> <p>1. Your UNIX-The Ultimate Guide, Sumitabha Das, Tata Mc GrawHill,</p>	
<p>Reference Books:</p> <p>1. "Unix Shell Programming", Yashwant Kanetkar, 2. "Beginning Shell Scripting", Eric Foster-Johnson, John C Welch, Micah Anderson, Wrox publication. 3. UNIX: Concepts and Applications, Sumitabha Das, Tata Mc GrawHill,</p>	

Web Programming [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA13	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Develop Web applications using XHTML and various scripting languages.</p> <p>CO 2: Build dynamic documents using Document Object Model (DOM).</p> <p>CO 3: Design documents using markup languages and stylesheets.</p> <p>CO 4: Justify the need for CGI programming between PERL and various mark-up Languages</p> <p>CO 5: Build the ability to select the essential technology needed to develop and implement web Applications.</p>			
Modules			Teaching Hours
Module 1			10 Hours
<p>Introduction to XHTML; Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links. Lists, Tables, Forms, Frames.</p> <p>Cascading Style Sheets: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The box model, Background images, The and <div> tags, Conflict resolution.</p> <p>The Basics of JavaScript: Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts,</p>			
Module 2			10 Hours
<p>Java Script and XHTML Documents The JavaScript Execution Environment, The Document Object Model, Elements Access in Java Script, Events and Event Handling, Handling Events from Body Elements, Handling Events from Text Box and password Elements, The DOM2 Event Model, The navigator Object, Dom Tree Traversal and Modification.</p> <p>Dynamic Documents with JavaScript: Introduction, Positioning Elements, Moving Elements, Element Visibility, Changing Colors and Fonts, Dynamic Content, Stacking Elements, Locating the Mouse Cursor, Reacting to a Mouse Click, Slow Movement of Elements, Dragging and Dropping Elements</p>			
Module 3			10 Hours
<p>Introduction to XML Introduction, Syntax, Document Structure, Document type definitions, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors, Web services.</p>			

Module 4	10 Hours
<p>Perl and CGI Programming Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples. Using Perl for CGI Programming: The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module. Building Web Applications with Perl Uploading files, Using Relational Databases.</p>	
Module 5	10 Hours
<p>4. Introduction to PHP Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements, Arrays, Functions, Pattern matching, Form handling, Files Building Web applications with PHP Tracking users, cookies, sessions, Using Databases, Handling XML.</p>	
<p>Question paper pattern:</p> <ul style="list-style-type: none"> · The question paper will have ten questions. · Each full question consists of 20 marks. · There will be 2 questions from each module. · Each question will have questions covering all the topics under a module. <p>The students will have to answer 5 full questions, selecting one full question from each module.</p>	
<p>Text Books:</p> <p>1. Robert W. Sebesta: Programming the World Wide Web, 4thEdition, Pearson education, 2012.Chapters 2, 3, 4, 5, 6, 7, 8, 9, 11&13</p>	

Computer Organization [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA14	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Understand the Basics of Digital System.</p> <p>CO2: Understand the Basics of Computer System Organization.</p> <p>CO3: Understand the concepts of the number system in Designing Digital System.</p> <p>CO4: Gain knowledge on combinational circuits and sequential circuits.</p> <p>CO5: Analyse the need of Logic circuits in digital system.</p>			
Modules			Teaching Hours
Module 1			10 Hours
<p>Binary Systems and Combinational Logic Digital Computers and Digital Systems, Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, subtraction using r's and (r-1)'s complements, Binary Code ,Binary Storage and Registers, Binary Logic. Axiomatic Definition of Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, The map Method, Two–and Three– Variable Maps, Four–Variables Map, Don't Care Conditions.</p>			
Module 2			10 Hours
<p>Arithmetic Circuits Digital Logic Gates, NAND and NOR Implementation, Introduction, Adders, Subtractors, Binary Parallel Adder, Decimal Adder, Magnitude Comparator, Decoders, Multiplexers, BOOTH algorithm for signed numbers with example.</p>			
Module 3			10 Hours
<p>Sequential Logic Introduction, different types of Flip– Flops, Triggering of Flip- Flops, Registers, Shift Registers, Ripple counter and Synchronous Counter. Machine Instruction: Introduction to Assemblers and Compilers, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.</p>			
Module 4			10 Hours
<p>Assembly Language and Input/ Output Organization Basics of Assembly Language Programme, Examples from Assembly Language Programming. Accessing I/O Devices, Interrupts, DMA, Processor Example, Buses.</p>			
Module 5			10 Hours
<p>The Memory System Basic Concepts, Semiconductor RAM Memories, Read– Only Memories, Speed, Size, and Cost, Cache Memories, Virtual Memories, Memory Management Requirements, Secondary Storage.</p>			
<p>Question paper pattern:</p> <ul style="list-style-type: none"> · The question paper will have ten questions. · Each full question consists of 20 marks. 			

- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. M. Morris Mano, "Digital Logic and Computer Design", Pearson, 2012.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th edition, Tata Mc Graw-Hill, 2011

Reference Books:

1. John P. Hayes, "Computer Architecture and Organization", Tata Mc Graw- Hill, Edition, 2012.
2. Soumitrs Kumar Mandal, "Digital Electronics Principles and Applications", Tata Mc Graw-Hill, 2010
3. Hamacher, " Computer Organization" , McGraw-Hill Education

Professional Communication and management [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA15	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<p>Course Outcome (CO): At the end of this course, the students will be able to CO1: Understanding the process of communication and its importance. CO2: Understanding how to write effective reports, Resumes, Letters, etc. CO3: Apply effective presentation strategies and group communication. CO4: Analyze motivation and leadership qualities. CO5: Understand the importance of ethics as an IT professional.</p>			
Modules			Teaching Hours
Module 1			10 Hours
<p>Communication in the workplace- Role of communication in Business, Process of Human Communication. Informal Communication- Listening, Non verbal Communication. Correctness Of Communication- Nature of correctness, Standard for Punctuation, Standards for Grammar, Standards for the use of numbers, Spelling, Capitalization.</p>			
Module 2			10 Hours
<p>Writing for Effect- Business Etiquette, conversational style, view point, positive language, and courtesy. Basics Report Writing- Defining Reports, Determine the Purpose, Determine the Factors, Gathering The Information, Organizing the Report, Writing The Report. Physical Presentation Of Letters, Memos, And Reports- Basics for all document preparation, Form of Business Letters. Strategies in the job search Process- Preparing application, Resume, Cover letter, Facing an Interview.</p>			
Module 3			10 Hours
<p>Group Communication- Introduction, Group discussion, Organizational group discussion, group discussion as part of selection process meetings, conferences. Effective Presentation strategies- Introduction, Defining purpose, Analyzing audience and locale, organizing contents, preparing outline, Visual Aids, Understanding Nuances of delivery, Kinesics, Proxemics, Paralinguistics, Chronemics, Sample speech.</p>			
Module 4			10 Hours
<p>Motivation- Motivation and Motivators, Motivation: The Carrot and the stick, The Hierarchy of needs Theory, The Motivation – Hygiene Approach to Motivation. Leadership- Defining Leadership, Ingredients of Leadership, Triat Approaches to Leadership, Leadership behavior and styles, Situational or contingency, Approaches to Leadership.</p>			
Module 5			10 Hours
<p>Ethics: An overview of Ethics- What are Ethics? Ethics in the business World, Ethics in Information Technology (IT).</p>			

Ethics for IT Professionals and IT users- IT professionals, The Ethical behavior of IT professionals, IT Users.	
--	--

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 20 marks.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

Text Books

1. Meenakshi Raman and Sangeeta Sharma: Technical Communication-Principles and Practices, Oxford University Press, 2004.
2. George Reynolds: Ethics in Information Technology, 2nd Edition, Thomson Course Technology, 2007
3. Lesikar and Flatley: Communication-Basic Business Communication Skills for Empowering the Internet Generation 9th Edition, Tata McGraw-Hill Edition.
4. Harold Koonlz and Heinz Weihrich: Management-Essentials of Management, 5th Edition, McGraw-Hill International Edition.

C Laboratory [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA16	CIE Marks	50
Number of Lecture Hours/Week	01 Hr Tutorial(Instructions) 02 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
CREDITS – 1.5			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Apply C concepts to develop interactive applications in C.</p> <p>CO2: Be fluent in the use of input output statements, constants, variables ,expressions and functions</p> <p>CO3: Be fluent in the use control structures, arrays ,structures and pointers</p> <p>CO4: Be fluent in the use of file handling techniques.</p>			
Laboratory Experiments:			
SECTION A			
<p>1. a. Check whether the given number is perfect number Defn: A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Example - The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$.</p> <p>b. Solve quadratic equations for the given values of a,b,c.</p>			
<p>2. Write a menu driven C program to</p> <p>a. Insert an element into an array</p> <p>b. Delete an element from the array (first occurrence)</p>			
<p>3. Write a C program to find the saddle point of a matrix.</p> <p>Defn: Given a RxC Matrix, A, i.e. R rows and C columns we define a Saddle-Point as Saddle_Pt (A) for a row I and column j is that A(i,j) that is the minimum of Row i and the maximum of Col j.</p>			
<p>4. Write a Menu driven C program to</p> <p>a. Accept two numbers n and m</p> <p>b. Sum of all integers ranging from n to m</p> <p>c. Sum of all odd integers ranging from n to m</p> <p>d. Sum of all even integers ranging from n to m</p> <p>Display an error message if $n > m$. Create functions for each of the options</p>			
<p>5. Write a Menu Driven C Program to implement the following using recursion</p> <p>a. Factorial of a number</p> <p>b. Fibonacci series</p>			
<p>6. Create a structure Complex Number having real and imaginary part as properties. Write functions to add and subtract the two complex numbers.</p>			
<p>7. Write a menu driven C Program</p> <p>a .to copy two strings</p> <p>b. to compare two strings</p> <p>c. to reverse a string</p> <p>using pointers and not using any library functions.</p>			

b. Write a C Program to compare two strings without using library function. Make use of pointers.

8. a. Write a C Program to remove all white spaces and newline characters from a file.
b. Find whether a given word exists in the file. If it exists display the location of the word

SECTION B

Develop a project using the C language and concepts learnt in the theory and exercises listed in part A with a good look and feel effects.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report that must include the following
 - a. Source Code
 - b. Sample output
4. The report must be evaluated for 5 Marks. Demonstration and Viva for 15 Marks.

Instructions:

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to Three students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

Sample suggestion for the Project:

Student Database project: Define a structure called student having the properties of student_id, student name and branch of the student with a sub structure of marks of 3 subjects. Write a

Menu Driven C Program to

- a. Add new student detail
- b. Delete a student detail
- c. Display all student details
- d. Display the name of the student with the best mark
- e. Display the name of the student with the worst mark
- f. Display the average marks scored by the students

Note: In the examination each student should do one question out of the above 8 questions

Web Designing Laboratory [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA18	CIE Marks	50
Number of Lecture Hours/Week	01Hr Tutorial(Instructions)	SEE Marks	50
	02 Hrs Laboratory		
		SEE Hours	03
CREDITS – 1.5			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Understand the concept and usages web based programming techniques.</p> <p>CO2: Be fluent in the use of XHTML programs using Javascript.</p> <p>CO3: Be fluent in the use of CGI and Perl programs for different types of server side applications.</p> <p>CO4: Be fluent in developing Web Applications using PHP.</p> <p>CO5: Design and implement user interactive dynamic web based applications.</p>			
<i>PART –A (Laboratory Experiments)</i>			
<p>1a) Develop and demonstrate a XHTML file that includes JavaScript script for the following problems:</p> <p>i) Input: A number n obtained using prompt Output: The first n Fibonacci numbers</p> <p>ii) Input: A number n obtained using prompt Output: A table of numbers from 1 to n and their squares using alert</p> <p>1b) Develop and demonstrate using JavaScript, a XHTML document that displays random numbers (integers).</p>			
<p>2a) Develop and demonstrate, using JavaScript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two uppercase characters followed by two digits followed by two uppercase characters followed by three digits; No embedded spaces allowed) of the user. Event handler must be included for the Form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.</p> <p>b) Modify the above program to get the current semester also (restricted to be a number from 1 to 6).</p>			
<p>3a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.</p> <p>b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.</p>			
<p>4a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.</p> <p>4b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.</p>			
<p>5a) Write a PERL program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.</p> <p>5b) Create XHTML form with Name, address line1, address line2 and email text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on name.</p>			

6. Write a PHP program to read student data from an XML file and store into the MySQL database. Retrieve and display using SEARCH function.

Part-B

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report (25-30 pages) that must include the following
 - a. Introduction
 - b. Requirement Analysis
 - c. Software Requirement Specification
 - d. Analysis and Design
 - e. Implementation
 - f. Testing
4. The report must be evaluated for 10 Marks. Demonstration and Viva will be evaluated for 10 Marks.

Instructions:

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to 2 or 3 students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

Note: In the examination each student should choose one out of the 6 questions from PART-A

UNIX Programming Laboratory [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	16MCA17	CIE Marks	50
Number of Lecture Hours/Week	01 Hr Tutorial(Instructions) 02 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
CREDITS – 1.5			
<p>Course Outcome (CO): At the end of this course, the students will be able to</p> <p>CO1: Understand the Unix programming environment.</p> <p>CO2: Be fluent in the use of Vi editor.</p> <p>CO3: Be able to design and implement shell scripts to manage users with different types of permission and file based applications.</p> <p>CO4: Be fluent to write Awk scripts.</p>			
<i>Laboratory Experiments:</i>		PART - A	
<p>A. Explore the Unix environment and Explore vi editor with vim tutor. Perform the following operations using vi editor, but not limited to:</p> <ol style="list-style-type: none"> 1. Insert character, delete character, replace character 2. save the file and continue working 3. save the file and exit the editor 4. quit the editor 5. quit without saving the file 6. rename a file 7. insert lines, delete lines, 8. setline numbers 9. search for a pattern 10. move forward and backward 			
<p>1. Write a shell script that takes a valid directory name as an argument and recursively descend all the subdirectories, finds the maximum length of any file in that hierarchy and writes this maximum value to the standard output.</p>			
<p>2. Write shell script to implement terminal locking (similar to the lock command). It should prompt the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if match occurs, it must lock the keyword until a matching password is entered again by the user, Note that the script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.</p>			
<p>3. Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case, the starting directory as well as all its sub directories at all levels must be searched. The script need not include any error checking.</p>			
<p>4. Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits</p>			
<p>5. Write a shell script to list all the files in a directory whose filename is at least 10 characters. (us expr command to check the length)</p>			

6. Write a shell script that accept a list of filenames as its argument, count and report occurrence of each word that is present in the first argument file on other Argument files.

7. Write a shell script that reports the logging in of a specified user within one minute after he/she login. The script automatically terminate if specified user does not login during a specified period of time.

8. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th, a “\” is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be supplied through a text file created by the user.

9. Write an awk script to delete duplicated line from a text file. The order of The original lines must remain unchanged.

10. Write an awk script to compute gross salary of an employee accordingly to rule given below. If basic salary is < 10000 then HRA=15% of Basic & DA=45% of basic. If basic salary is >= 10000 then HRA=20% of basic & DA=50% of basic.

PART- B

Develop a project using the Unix concepts learnt in the theory and exercises listed in part A with a good look and feel effects.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report that must include the following
 - a. Source Code
 - b. Sample output
4. The report must be evaluated for 5 Marks. Demonstration and Viva for 15 Marks.

Instructions:

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to Three students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

Sample suggestion for the Project:

Developing Shells

Developing Editors with all basic properties

Note: In the examination each student should choose one out of the 10 questions from PART-A