Course Code: 22BS102/152

ENGINEERING CHEMISTRY

(Common to all Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives: By studying this course students

1. Are exposed to the importance of water and its treatment for domestic and industrial usage.
2. Get adaptability to new developments in fundamental aspects of batteries, the significance of corrosion its control to protect the structures.
3. Get an awareness of the chemistry of polymers and their engineering applications.
4. Learn the basic concepts of petroleum and its products.
5. Acquire required knowledge about engineering materials like lubricants, refractories, and smart materials.

Unit I – Water and its treatment

Introduction to the hardness of water, types of hardness, causes of hardness, Expression of hardness- Estimation of hardness of water by the complexometric method using EDTA. Numerical problems on the hardness of water and complexometry.

Boiler troubles: Sludges, Scales, and Caustic embrittlement (definition, cause, effect, and removal). External treatment methods - Softening of water by ion-exchange processes. Desalination of water-Reverse osmosis.

Sewage Water - Biological oxygen demand (BOD), Chemical oxygen demand(COD) - Determination and significance.

Potable water: its characteristics and processing-Disinfection of potable water by chlorination (break-point chlorination), Ozonation.

Unit II – Batteries & Corrosion

Batteries: Introduction - Classification of batteries- primary and secondary batteries with examples. Construction, working, and applications of Lithium cells with the solid cathode (Li-MnO_2 cell), Zn-air, and Lithium-ion battery, applications of Li-ion battery to electrical vehicles. Fuel Cells- Differences between a battery and a fuel cell, Construction, and applications of Polymer Electrolyte Membrane Fuel Cell (PEMFC), direct methanol fuel cell (DMFC).

Corrosion: Causes and effects of corrosion – theories of chemical and electrochemical corrosion -mechanism of electrochemical corrosion. Factors affecting the rate of corrosion: position of the metal in galvanic series, nature of corrosion product (Pilling-Bedworth ratio & rule), purity of metal, the effect of temperature, and effect of pH. Corrosion control methods- Cathodic protection of Iron – Sacrificial anode and impressed current methods.

Unit III – Polymeric materials

Introduction, Definitions of monomer, polymer, functionality, and degree of polymerization-Classification of polymers with examples-Types of polymerization–addition (mechanism of free radical addition polymerization) polymerization of polyethylene, polyvinyl chloride, Polystyrene and condensation polymerization of Nylon 6:6

Plastics: Definition and characteristics- thermoplastic and thermosetting plastics, Preparation and engineering applications of Teflon and Terylene,

The molecular mass of a polymer: Number average molecular mass method and weight average molecular mass method-Numerical problems.

Conducting polymers: Classification-Preparation-mechanism of conduction in trans-polyacetylene and applications of conducting polymers.

Biodegradable polymers: Concept and advantages - Polylactic acid and polyvinyl acetate and their applications.

Unit IV – Energy Sources

Introduction, Calorific value of fuel – HCV, LCV- Dulong's formula, and numerical problems.

Solid fuels: Analysis of coal–proximate and ultimate analysis with their significance.

Liquid fuels: Petroleum and its refining, Cracking: Types-thermal cracking, catalytic cracking-moving bed catalytic cracking. Knocking – octane and cetane number, synthetic petrol - Fischer-Tropsch's process. 2G-ethanol-preparation from renewable sources and applications.

Gaseous fuels: Composition and uses of Natural gas, LPG, and CNG

Unit V – Engineering Materials

Lubricants: Classification of lubricants with examples-characteristics of a good lubricant - mechanism of lubrication (thick film, thin film, and extreme pressure). Properties of lubricants: viscosity, cloud point, pour point, flash point, and fire point (Determination and their significance).

Smart materials and their engineering applications: Introduction, classification, and applications.

Course Outcomes: On completing the course a student will be able to

CO 1 : Relate the basic properties of water and its usage for domestic and industrial purposes.
CO 2 : Summarize the basic knowledge of electrochemical procedures related to batteries and corrosion and its control.
CO 3 : Apply the fundamentals and general properties of polymers and other engineering materials.
CO 4 : Analyze real-time situations related fuel energy sources.
CO 5 : Predict potential applications of chemistry and the practical utility of engineering materials in order to become good engineers and entrepreneurs.

Textbooks:

1. Engineering Chemistry by Rama Devi, and Rath, Cengage Publications, 2022.
2. A textbook of Engineering Chemistry by M.Thirumala Chary, E. Laxminarayana and K. Shashikala, Pearson Publications, 2021.
3. Textbook of Engineering Chemistry by Jaya Shree Anireddy, Wiley Publications.

References:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010.
2. Engineering Chemistry by Shikha Agarwal, Cambridge University Press, Delhi (2015).
3. Engineering Chemistry by Shashi Chawla, Dhanpatrai and Company (P) Ltd. Delhi (2011).

Course Code: 22BS103

MATHEMATICS FOR COMPUTING

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS and IT Branches)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To study the first order Ordinary differential equations and acquire the skill of finding analytical solutions of such equations
2. To study the higher order Ordinary differential equations and Difference equations and to acquire the skill of finding solutions of such equations and to use them in engineering applications
3. To understand the geometrical approach to the Mean value theorems and their applications to the mathematical problems. Evaluation of improper integrals using Beta and Gamma functions.
4. To understand the concept of partial derivative, total derivative and to use them in finding the extreme values of a multi-variate function with/without constraints.
5. To identify the nature of a series using the appropriate test for convergence.

Unit I - First order Ordinary Differential Equations

Exact differential equations, Equations reducible to exact differential equations, Linear and Bernoulli’s equations, Applications: Newton’s law of cooling, Law of natural growth and decay. System of linear Ordinary Differential Equations.

Unit II - Higher order Ordinary Differential Equations and Difference Equations

Linear differential equations with constant coefficients: Non-Homogeneous terms of the type e^{ax}, $\sin ax$, $\cos ax$, polynomials in x, $e^{ax}V(x)$ and $x\,V(x)$, method of variation of parameters.

Difference Equations: Order and Degree-Linear Difference equations with constant coefficients-Complementary Function-Particular Integrals of the types a^n, polynomial in n.

Unit III - Uni-variate Calculus

Mean value theorems: Rolle’s theorem, Lagrange’s Mean value theorem (without proofs) with their Geometrical Interpretation and applications, Taylor’s Series. Beta and Gamma functions and their applications. Fourier series over a general interval $[C, C+2L]$.

Unit IV - Multi-variate Calculus

Calculus: Partial differentiation, Total derivatives, Chain rule, Jacobian, Hessian. Functional dependence & independence, Maxima and minima of functions of two and three variables, Method of Lagrange multipliers.

Unit V - Sequences and Series

Sequences: Definition of a Sequence, Limit, Convergent, Divergent and Oscillatory sequences. Series: Convergent, Divergent and Oscillatory Series- Series of positive terms- Comparison test, Root test, Ratio test, Raabe’s test. Alternating series-Leibnitz test. Absolute and Conditional Convergence.

Course Outcomes: At the end of the course, the student should be able to

CO 1	:	Solve the first-order Ordinary Differential Equations and extend the knowledge to the applications in engineering problems.
CO 2	:	Solve higher-order Ordinary Differential Equations and Difference equations and extend the knowledge to the applications in engineering problems.
CO 3	:	Apply Mean value theorems to solve engineering problems and to evaluate improper integrals using Beta and Gamma functions.
CO 4	:	Find the extremum of a multi-variate function with/without constraints.
CO 5	:	Determine the convergence/divergence of a given infinite series.

Textbooks:

1. Higher Engineering Mathematics, B.S. Grewal, 36th Edition, Khanna Publishers, 2010.
2. Higher Engineering Mathematics, H. K. Dass and Er. Rajnish Verma, S Chand and Company Limited, New Delhi.

References:

1. Advanced Engineering Mathematics, Erwin Kreyszig, 9th Edition, John Wiley & Sons, 2006.
2. Calculus and Analytic geometry, G.B. Thomas and R.L. Finney, 9th Edition, Pearson, Reprint, 2002.
3. A textbook of Engineering Mathematics, N.P. Bali and Manish Goyal, Laxmi Publications, Reprint, 2008.
4. Advanced Engineering Mathematics, R.K. Jain and S.R.K. Iyengar, 5th Edition, Narosa Publications, 2016.

Course Code: 22CS101

PROBLEM SOLVING THROUGH ‘C’

(Common to all Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To introduce student to the fundamental concepts of C programming, structured constructs and syntax.
2. To enable student to formulate simple algorithms for solving arithmetic and logical problems.
3. To familiarize students with modular programming in implementing solutions for complex problems.
4. To enable student to apply appropriate concepts like pointers, arrays for a particular algorithm implementation.

Unit I – Introductory Concepts

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.), Types of Programming Languages.

Idea of Algorithm: Steps to solve logical and numerical problems.

Representation of Algorithm: Flowchart/Pseudo code with some conceptual examples and exercises.

From algorithms to programs Creating and Running Programs, Syntax and Logical Errors in compilation, object and executable code.

Introductory Concepts: Introduction to C, Simple C Programs, Desirable Program Characteristics.

C Fundamentals: The C Character Set, Identifiers and Keywords, Data Types, Constants and Variables, and Declarations.

Operators and Expressions: Arithmetic Operators, Unary Operators, Relational and Logical Operators, Assignment Operators, Bitwise Operators, the Conditional Operator, Math library functions.

Data Input and Output: Single Character I/O functions-getchar, putchar, I/O statements-scanf, printf, gets, puts functions.

Unit II – Programming Constructs

Control Statements: Selection Statements:2-way selection (if, nested if, if-else), multi-way selection (else-if ladder, switch-case), break, continue statements**.**

Iterative Statements: Pretest Loops (for, while), post-test loops (do-while)

Functions: Function – Prototype, Definition, call, Passing arguments to a function, Example programs.

Scope and Extent: Local and global scope, extent, Storage Classes: Automatic, Extern, Static, Register.

Unit III – Recursion, Arrays

Recursion: Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, GCD etc.

Preprocessor Directives: File Inclusion, Macros.

Arrays: Defining an array, processing an array, passing arrays to functions.

Multidimensional Arrays: Example programs on matrix operations

Unit IV – Pointers, Strings

Pointers: Pointer Declarations, Passing pointers to functions, NULL pointer, Pointers and one-dimensional Arrays, Dynamic memory allocation, operations on pointers, pointers and multidimensional arrays, arrays of pointers.

Strings: String manipulation using user defined and library functions (string.h, ctype.h)

Unit V – Searching and Sorting

Searching: Linear Search, Binary Search – Iterative and Recursive implementations
Sorting: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort Algorithms.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Understand programming concepts and analyze a problem, design a solution and develop an algorithm to solve it.

CO 2 : Modularize a problem and implement the solution using basic programming concepts, control statements and functions.

CO 3 : Evaluate the use of macros and implement solutions to complex problems using recursion and homogeneous data types.

CO 4 : Implement solution using pointers for problems of relevance and use different dynamic memory allocation methods.

CO 5 : Understand and analyze, differentiate and implement elementary algorithms of sorting, searching and will also be able to compare and contrast algorithms with respect to time and space complexity.

Textbooks:

1. Programming with C (Schaum's Outlines Series), Byron S. Gottfried, 3rd Edition, McGraw-Hill, 2017.
2. Programming with C, Ajay Mittal, 9th Impression, Pearson Education Ltd, 2017.

References:

1. The C Programming Language, Brian W. Kernighan and Dennis M. Ritchie, 2nd Edition, Prentice Hall of India, 1998.
2. C Programming & Data Structures, B.A.Forouzan and R.F. Gilberg, 3rd Edition, Cengage Learning, 2010.

Course Code: 22ME101/151

COMPUTER AIDED ENGINEERING DRAWING

(Common to CSE, AIML, CS, DS, IT, ECE, EEE & EIE Branches)

Instruction	:	1 Period/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	4 Periods/week	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand Standards conventions and use AutoCAD commands for drawing various geometrical constructions and curves used in engineering practice.
2. To acquire skills to solve problems on the orthographic projection of points and lines.
3. To understand the orthographic projection of planes and solids.
4. To understand section of solids and the development of surfaces.
5. To grasp the concept of converting isometric projection to orthographic projection and vice versa.

Unit I

Introduction to AutoCAD Software – The Menu System, Toolbars (Standard, Object Properties, Draw, Modify and Dimension), Drawing Area (Background, Crosshairs, Coordinate System), Dialog boxes and windows, Shortcut menus (Button Bars), The Command Line, The Status Bar, Different methods of zoom as used in CAD, Select and erase objects.

Conventions in Drawing – BIS Conventions. Lettering, Dimensioning.

Engineering Curves – Construction of Ellipse, Parabola, and Hyperbola – General method and Cycloidal Curves – Cycloid, Epi, and Hypo Cycloids.

Unit II

Principles of Orthographic Projections – Conventions – Fundamentals of First and Third Angle projections, Projections of Points

Projection of Lines – Line is parallel to both planes, Line is parallel to one and perpendicular to the other, Line is inclined to one plane and parallel to another plane, Line is inclined to both planes.

Unit III

Projection of Planes - Projections of regular Planes – planes parallel to one and perpendicular to another plane, planes perpendicular to one and inclined to the other, planes inclined to both planes.

Projection of solids: Projections of Regular Solids - Cone, Cylinder, Prism, Pyramid – Axis is parallel to one and perpendicular to other, Axis is inclined to one and parallel to other.

Unit IV

Sections of Right Regular Solids - Cone, Cylinder, Prism, and Pyramid – Sectional plane parallel to one plane and perpendicular to the other and sectional plane inclined to HP and perpendicular to VP.

Development of Surfaces of Right Regular Solids – Cone, Cylinder, Prism, and Pyramid - Sectional plane parallel to one and perpendicular to the other and sectional plane inclined to HP and perpendicular to VP.

Unit V

Principles of Isometric Projection – Isometric Scale – Isometric Views, Conventions, Isometric Projections and Views of simple Plane figures **–** Regular Polygons and circle. Isometric Projections and Views of simple solids – Prism, Pyramid, Cylinder, and Cone.

Conversion of Isometric Views to Orthographic Views – Drawing of Front, Top and Side views from isometric views of objects.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Know the Standard conventions and Construction of various Engineering curves through Auto CAD.

CO 2 : Apply fundamentals of the theory of projections and draw orthographic projections of points and lines in any position through Auto CAD.

CO 3 : Construct orthographic projections of simple planes and regular solids in any position through Auto CAD.

CO 4 : Draw sectional views and developments of various basic 3D objects through Auto CAD.

CO 5 : Construct isometric views and construct multi-view drawings of simple 3D objects through Auto CAD.

Textbooks:

1. Engineering Drawing, Basant Agarwal, C M Agarwal, 3rd Edition, McGraw-Hill, 2020.
2. Engineering Graphics with AutoCAD, D. M. Kulkarni, A. P. Rastogi, A. K. Sarkar, Revised Edition, PHI Learning Pvt., Ltd., 2009.

References:

1. Engineering Drawing, N.D. Bhat, 53rd Edition, Charotar Publishers, 2016.
2. Engineering Drawing and Graphics, Venugopal, 5th Edition, New age Publishers, 2020.

Course Code: 22CS102

ESSENTIALS OF SYSTEM AND WEB INTERFACING

(Common to CSE, CSE-AI&ML, CSE-CS and CSE-DS Branches)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To offer exposure on computer system interfacing and pre-processing data using filters
2. To develop proficiency in students for developing efficient shell scripts using constructs
3. To explore the CMS tools like WordPress and make a blog post
4. To make students understand the importance of good web interface design
5. To enable students to become conversant with styling constructs of CSS

Unit I

Computer System: Introduction, Defining Computer, Exploring the Basic Parts of a Computer, Describing Computer Hardware, Describing Computer Software.

Exploring an Operating System- Introduction to functions of Operating System, Introduction to UNIX, Why UNIX, Architecture.

UNIX SHELL Introduction: Introduction to UNIX Shell, simple commands, Syntax of commands, uname and bc commands, variables, and quotes. Filesystem layout. File-related commands, Types of files, links- hard and soft links, Operations on Directories. Security and permissions. Redirection. Job control- ps, fg and bg.

Unit II

Simple filters: filters and pipes, concatenating files, displaying the beginning and end of files, cutting, sorting, translating characters, wc, comparing files using diff, comm.

Filters using regular expressions: patterns, regular expressions, grep family, regular expressions supported by grep family, searching based on content.

Unit III

Communications: Introduction to Remoting, Configuring FTP and Telnet servers

Korn shell programming: Environment and shell variables, basic script concepts, expressions, decisions, making selections, repetition, special parameters, and variables, changing positional parameters, argument validation, debugging scripts, and script examples.

Unit IV

Introduction to CMS WordPress: Introduction to CMS, Introduction to WordPress, Web Publishing for masses, how WordPress works, the lifecycle of Word Press blog post, Download WordPress and Install, uploading your WordPress files to the web server, Working with Hosting panel, Using the Dashboard, and its components, create a first blog post, preview and publish your post.

Introduction to HTML

Document Structure, Basic formatting elements, links and navigation, image, image maps, List, Tables, and Forms. HTML 5: semantic elements, Embedding Media (video and audio), storage: local, session.

Unit V

Introduction to CSS: Style and link tags, selectors and its types, box model, positioning, styles: background, list, border, padding, margin. CSS 3: Responsive design using media queries, flex, grid, transitions and animations.

Course Outcomes: At the end of the course, the student should be able to

CO 1	:	Interact with the UNIX system modules and utilize utilities
CO 2	:	Apply file filters and generate analytics using awk and grep tools
CO 3	:	Develop efficient shell scripts using bash constructs.
CO 4	:	Develop a simple web page using HTML elements.
CO 5	:	Present the content with good user experiences and make pages responsive.

Textbooks:

1. Unix and Shell Programming, Behrouz A. Forouzan, Richard F. Gilberg. Thomson, 2012 Brooks/Cole Publishing, 2003.
2. Beginning HTML, XHTML, CSS and JavaScript, Jon Ducket, Wiley India Pvt. Ltd, Wrox Publication, 2010 and reprint 2018.

References:

1. Head First WordPress, Jeff Sairto, O’Reilly Media, Inc., First Edition, 2010.
2. Responsive Web Design with HTML5 and CSS 3, Ben Frain, 3rd Edition, Packt Publication, 2012.
3. Unix for programmer and users, 3rd edition, Graham Glass, King Ables, Pearson Education, 2003.
4. Unix Programming environment, Kernighan and Pike, PHI/Pearson Education,1984.
5. Computer Science: An Overview, Glenn Brookshear & Dennis Brylow, 12th Edition, Pearson Education Limited, 2018.

Course Code: 22BS132/182

ENGINEERING CHEMISTRY LAB

(Common to all Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives: The course consists of experiments related to the principles of chemistry required for an engineering student. It is aimed to train the students

1. In estimating the chemical substances by a set of procedures involving titrimetric analysis.
2. To expose the students to various instruments such as conductometer, potentiometer, pH meter, colorimeter, and viscometer.
3. In different techniques are involved in the qualitative and quantitative analysis of substances.
4. To learn preparation and identification techniques of a polymer in the laboratory.
5. To make appropriate measurements, analyze the data and report the results.

List of experiments: (any 10 of the following)

1. Estimation of the Hardness of water by EDTA Complexometry method.
2. Estimation of the concentration of strong acid by Conductometry.
3. Estimation of the concentration of the mixture of acids by Conductometry
4. Estimation of the amount of Fe^{2+} ion by Potentiometry.
5. Estimation of the concentration of strong acid by Potentiometry.
6. Determination of concentration of acid by pH meter.
7. Determination of alkalinity of water.
8. Preparation of sanitizer.
9. Preparation of a polymer: Polystyrene.
10. Determination of viscosity by using a Redwood Viscometer.
11. Determination of the rate of corrosion of mild steel in the presence and absence of inhibitor
12. Verification of Beer's law using $CuSO_4$ solution by Colorimetry.
13. Saponification value of coconut oil.

Course Outcomes: At the end of the course a student will be able to

CO 1	:	determine the parameters like the hardness of water, alkalinity, and rate of corrosion of mild steel
CO 2	:	estimate the acid concentration by conductometry.
CO 3	:	analyze instrumental techniques such as potentiometry and pH meter in order to find out the concentrations or equivalence points.
CO 4	:	interpret molecular/system properties such as viscosity, and saponification value of coconut oil.
CO 5	:	apply analytical skills about colorimeter/ polymer/Sanitizer.

Textbooks:

1. Lab manual for Engineering chemistry Ramadevi and Aparna 2022 S. Chand Publications.
2. Vogel's textbook of practical organic chemistry 5th Edition College.
3. Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007).

References:

1. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.
2. Instrumental methods of Chemical Analysis, Chatwal, Anand, Himalaya publications.

Course Code: 22HS131/181

ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

(Common to all Branches)

Instruction & Activity	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
			Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives: The main aim of this course is

1. To train the listener to capture content accurately.
2. To speak fluently and appropriately in a neutral accent in the given context.
3. To encourage the students to read extensively to develop productive skills.
4. To write with precision in different contexts, for a variety of purposes and employ appropriate styles.
5. To empower students with proficiency in LSRW Skills of English.

SYLLABUS

1. Introduction to LSRW skills
2. Listening skills and Retelling a story
3. Introduction to Consonants and Vowels
4. Transcription, Syllable, Syllabic Division and Syllable Stress
5. Intonation and Semantic Implication
6. Brief Speeches and Public Speaking
7. Non-Verbal Communication and Role Plays
8. Presentation Skills and Information Transfer
9. Group Discussion
10. Report Writing – Analytical and Informative
11. Picture Description

Learning Software

"K-VAN Solutions" and "English Grammar in Use" are used in practice sessions for the following topics:

Exercise – I

CALL Lab:

Understand: Listening Skill- Its importance – Purpose- Process- Types- Barriers- Effective Listening.

Practice: Introduction to Phonetics – Speech Sounds – Vowels and Consonants – Minimal Pairs- Consonant Clusters- Past Tense Marker and Plural Marker- Testing Exercises

ICS Lab:

Practice: Ice-Breaking Activity and Brief Speeches.

Exercise – II

CALL Lab:

Understand: Structure of Syllables – Word Stress– Weak Forms and Strong Forms – Stress pattern in sentences – Intonation.

Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms- Stress pattern Sentences – Intonation & Semantic Implications- Testing Exercises

ICS Lab:
Practice: Situational Dialogues – Role Play- Expressions in Various Situations –Making Requests and Seeking Permissions.

Exercise – III

CALL Lab:
Understand: Errors in Pronunciation - Neutralising Mother Tongue Interference (MTI),
Practice: Phonetic transcription - Common Indian Variants in Pronunciation – Differences between British and American Pronunciation -Testing Exercises

ICS Lab:
Practice- Narrations- Retelling a story, Picture Description

Exercise – IV

CALL Lab:
Understand: Listening for General Details.
Practice: Listening Comprehension Tests - Testing Exercises

ICS Lab:
Public Speaking – Exposure to Structured Talks - Non-verbal Communication - Presentation Skills.
Practice: Making a Short Speech – Extempore- Making a Presentation.

Exercise – V

CALL Lab:
Listening for Specific Details.
Practice: Listening Comprehension Tests -Testing Exercises

ICS Lab:
Group Discussion, Report Writing and Information Transfer
Practice: Group Discussion

Course Outcomes: At the end of the course, students will

CO 1	:	Comprehend and respond appropriately in various scenarios
CO 2	:	Emerge as confident and competent communicators of the English Language
CO 3	:	Apply pronunciation skills to evolve as proficient speakers
CO 4	:	Analyze and compose effectively across various mediums
CO 5	:	Develop critical and analytical thinking

References:

1. Balasubramanian, T. *A textbook of English phonetics for Indian students*. Macmillan, 1981.
2. Sethi, J., and Pushya Vibhooti Dhamija. *A course in phonetics and spoken English*. PHI Learning Pvt. Ltd., 1999.
3. Redman, Stuart, and Ruth Gairns. *Test Your English Vocabulary in Use*. Cambridge University Press, 2008.
4. Deo, Karan., *Group Discussion*, Ramesh Publishing House, 2013
5. Anderson, Marilyn, Pramod K. Nayar, and Madhuchanda Sen. *Critical Thinking, Academic Writing and Presentation Skills*. Dorling Kindersley, 2012.

Course Code: 22CS131

C PROGRAMMING LAB

(Common to all Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To formulate problems and implement algorithms using C programming language.
2. To introduce role of constants, variables, identifiers, operators, type conversion and other building blocks of C Language.
3. To impart role of functions involving the idea of modularity, usage of arrays, pointers, structures for developing solutions to complex problems.
4. To introduce programming using gcc compiler in Linux.

All the programs need to be implemented in GDB mode in a Linux Environment.

Task 1:

1. Write the algorithm and draw the flow chart to find the roots of a quadratic equation
2. Write the algorithm and draw the flow chart to find the sum of digits of a given n digit number.
3. Write a C program to explore decimal, octal, hexadecimal, unsigned, unsigned long long formats of integers with printf and scanf functions.
4. Write a C program to convert the given temperature in Celsius into Fahrenheit.

Task 2:

1. Write a simple calculator program which reads operand1, operator and operand2 as input and displays the result.
2. Write a C program to find the greatest of 2 numbers
3. Write a C program to find the greatest of 3 numbers
4. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Task 3:

1. A number is said to be Armstrong if the number is equivalent to the sum of cubes of its digits. Write a C program to check whether a given number is Armstrong or not.
2. Write a C program to find the sum of individual digits of a positive integer.

Task 4:

1. Write a C Program to generate the following pattern.

```
    1
   123
  12345
 1234567
123456789
 1234567
  12345
   123
    1
```

2. Write a C program to define the macros SUM (a, b), SQUARE (a) and SQUARE (SUM (a, b)) and print the results.

3. Write the calculator program which reads operand1, operator and operand2 as input and displays the result to execute different operations like addition, subtraction, multiplication, division etc. until user’s choice is exit.

Task 5:

1. Write a C program to illustrate functions without parameters and without return type, without parameters and with return type, with parameters and without return type and with parameters and with return type.
2. Write a C function to calculate the sine series sum 1- x3/3! + X5//5! and call the function.

Task 6:

1. Write a C program in which a recursive and non-recursive functions are called to compute factorial values based on user’s choice.
2. Write a C program in which a recursive and non-recursive functions are called to generate Fibonacci series based on user’s choice.

Task 7:

1. Write a C program to find the sum of the elements of a given list (array).
2. Implement two separate functions which return the minimum and maximum values of a given array-list and call these functions.

Task 8:

1. Write a C program to find the transpose of a given input matrix (read the dimensions of matrix too as input).
2. Implement two separate functions for finding the sum and product of matrices and call these functions.

Task 9:

1. Implement a C function to exchange the values of given two variables and call the function (using pointers).
2. Implement two separate C functions to perform insertion of an element and deletion of an element operations on an array at a specified position (pass the array and its size as pointers).
3. Write a C program to create a dynamic list of real numbers where the size of the list is accepted as input, extend its size and release it (use dynamic memory allocation functions).

Task 10:

1. Write a C program to accept string as input and find its length using a user-defined string length function, reverse the string and check whether the string is palindrome or not.
2. Implement a C function to read a multi-word string and copy the input string to other string (the destination string must be a dynamically allocated string).

Task 11:

1. Write a C program to implement Linear Search
2. Write a C Program to implement Binary Search

Task 12:

1. Write a C Program to implement Bubble Sort.
2. Write a C Program to implement Selection Sort.

Task 13:

1. Write a C Program to implement Insertion Sort.
2. Write a C Program to implement Quick Sort.

Task 14:

1. Write a C Program to implement Merge Sort.
2. Write a C Program to implement Linear Search and Binary Search using Recursion.

Task 15:

1. To print all strong numbers between given interval using functions.
   Note: Strong number is a special number whose sum of the factorial of digits is equal to the original number. For Example: 145 is strong number
2. Print Fibonacci series using recursion upto n numbers
3. Find LCM and GCD of two numbers using recursion.

Task 16:

1. Find the exponentiation of a number and the product of two numbers using recursion.
2. Given an integer N, print all the odd numbers from 1 to N in ascending order.
3. Given two integers A and B. Print all numbers from A to B inclusively, in ascending order, if A < B, or in descending order, if A ≥ B.
4. Write a C Program to solve the Towers of Hanoi Problem using recursion.

Task 17:

1. Write a program to print the frequency of each digit in a given integer.
2. Write a program to print the reverse of the number entered by the user.
3. Write a program to find whether a positive integer entered by the user is a palindrome or not.

Task 18:

1. Write a program to print the following pyramid or similar shapes for a user given positive n(<10). All the below sample outputs are for N=4

a)

```
1

1 2

1 2 3

1 2 3 4
```

b)

```
      1

    1 2 1

  1 2 3 2 1

1 2 3 4 3 2 1

  1 2 3 2 1

    1 2 1

      1
```

c)

```
      1
    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
```

Task 19:

1. Write a program to find 2’s complement of a number.
2. Write a program to find the sum of natural numbers using recursion.
3. Write a C program to remove duplicates in a given array and store it in a new array.
4. Write a C program to merge two arrays.
5. Write a C program to find number of elements in common between two given arrays.
6. Write a C program to swap the numbers when two adjacent elements in an array are odd numbers.
7. Write a C program to find whether an array is subset of another array

Task 20:

1. Write a C program to find a maximum occurring character in the input string using functions.
2. Write a C program to remove all duplicates from a given string.
3. Find the smallest window in a string containing all characters of another string
4. Write a program to revere words in a given string.

Note: Tasks 1 to 14 are mandatory and Tasks 15 to 20 are optional.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Familiarize with Linux programming environment and translate given algorithms to a working and correct program.

CO 2 : Interpret syntax errors as reported by the compilers and to be able to identify and correct logical errors encountered at run time using debuggers like GDB.

CO 3 : Write iterative as well as recursive programs.

CO 4 : Represent data in arrays, pointers, strings and manipulate them through a program.

CO 5 : Apply Algorithm for solving problems like sorting, searching.

References:

1. Schaum's Outline of Programming with C, Byron Gottfried, 2nd Edition, McGraw-Hill, 1996
2. Programming with C, Ajay Mittal, 9th Impression, Pearson Education Ltd, 2017.
3. The C Programming Language, Brian W. Kernighan and Dennis M. Ritchie, 2nd Edition, Prentice Hall of India, 1988.
4. C Programming & Data Structures, B.A.Forouzan & R.F. Gilberg, 3rd Edition, Cengage Learning, 2010.

Course Code: 22CS132

ESSENTIALS OF SYSTEM AND WEB INTERFACING LAB

(Common to CSE, CSE-AI&ML, CSE-CS and CSE-DS Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To facilitate the students in developing system interfaces and configure different services.
2. To enable students with a familiarity with developing content management system.
3. To offer hands-on experience in writing shell scripts to extract meaningful insights from text documentation.
4. To enable the students to present their ideas in an effective manner using office tools.

Task 1:

Working with the Installation of OS and disk partitioning

Install the Windows or Ubuntu OS in one of the machines provided in the laboratory with the support of the instructor. Do the disk partitioning and divide it into the required number of parts accordingly.

Task 2:

KORN Shell Commands and Scripting

1. Recording of shell session and carrying out the following tasks
   i. Display the kernel information b) Change terminal options for erase, end of file
   ii. Display the date in dd/mm/yyyy format.
2. Launch the g-editor tool and create few files
3. Create a regular file “Std-Details” with contents organized around the following format with tab space as a field separator.

   Name roll number % in SSC school-name district state

   Then complete the following tasks
   i. List the home directory and List the files under cwd in long format.
   ii. Concatenate /etc/passwd file and interpret its contents.
   iii. Display only two lines at a time from the contents of Std-Details using more and suppress multiple blank spaces
   iv. Create two subdirectories under the home directory and change to one of the subdirectories and create new files in it.
   v. Change to any sub-directory and create a few more files in it.
   vi. Copy any file present in the home directory into a subdirectory while specifying absolute and relative paths
   vii. Copy all C files from the home directory into a subdirectory using wild characters.
   viii. Create a hard link to any file in it and long list its attributes.
   ix. Create a soft link to any file and long list its attributes.

Task 3:

1. Create a regular file and change the access permissions using octal numbers and +/- options at different levels such as user, group, and other.
2. Create a new directory and disable execute permission on it and then try to change to that directory? If your attempt is unsuccessful, then explain the reason.
3. Find a file based on name, type and permission sequence
4. Apply relevant filter on Std-Details file to complete the following task
   i. Display the first 5 lines
   ii. Display the last 5 lines

iii. Display 5-8 lines
iv. Display the name of each students
v. Display the name and roll number of each student

5. Sort the file Std-Details based on % of marks field
6. Sort the file Std-Details based on % of marks and state in ascending and descending orders

Task 4:

1. Complete the following tasks using grep family
   i. Display all lines that start with Suresh
   ii. Display all blank lines
   iii. Display all students who born in the year 2000h
   iv. Display all the lines the lines that start and end with the same character
   v. Display all lines that start with 'S' and end with 'a'
   vi. Display the number of blank lines present in the file

2. Display the number of directories present under your home directory using pipe

Task 5:

Write shell scripts to accomplish the following tasks

i. To take command line arguments and display the number of arguments, a list of arguments
ii. Determine the type of the file and the access permissions set on a file that is passed as an argument
iii. Design a menu-driven shell program using select

Task 6:

Working with CMS – Word Press
Design a static blog post with three pages and publish it. Include bulleted text and images in each page. Name the pages as Home, About Us and Contact Us. Format the pages accordingly.

Task 7:

Working with HTML Basic Elements
Develop a static web page for personal Profile using the basic formatting elements of HTML. Use h1 to h6, list, paragraph, and table wherever necessary.

Task 8:

Working with form and anchor elements
Design Home Page, Catalog Page, About Page, Contact Us Page, Login Page, and Registration Page with static content (refer www.cvr.ac.in for content). Add a navigation bar at the top of the page. Upon clicking on the navigation link user must see the corresponding page output as response. Use various **<input>** elements to create the login and registration: text, password, checkbox, radio, select and option, text area. Use **<ul>** and **<li>** for creating navigation bar (either horizontal or vertical) and **<a>** for adding linking in the web page.

Task 9:

Working with the semantic elements of HTML 5
Use the various HTML 5 semantic elements like header, footer, main, section, aside and article and redo the experiment in Task 7.

Task 10:

CSS basic formatting.
Apply formatting to the Personal Profile page created in Task 6 using CSS rules. Background Properties, margin, border and padding properties (CSS Box Model), Text and Font properties.

Task 11:

Working with @mediaquery
Create multiple breakpoints and adjust the content to fit the device with using media queries.

Task 12:

Working with flex and grid layouts of CSS 3
Use the Task 7 and change the navigation bar to make use of the flex layout model while adding the navigation. Update the catalog page of Task 7 to use the grid and flex layout modules of CSS.

WORD PROCESSING

Task 13:

Preparing News Paper Article/Advertisement
Features to be covered:

- PAGE SETTINGS: Border, Background, Size, Layout, Numbers, Break, Header & Footer.
- TEXT FORMATTING: Color, Font, Size, Background, Border, Effects, Position.
- HEADINGS & INDEX, HYPERLINK.
- PARAGRAPH SETTINGS: 5 Alignments, Borders and shading, Inserting Special Symbols & Equations, Background, Drop cap
- IMAGE/CLIPARTS & DRAWING: Insert, Protect, Alignments, Transparency, Grouping.
- BULLETS: Types, Shapes & Symbols, Alignment.
- PAGE COLUMNS (IEEE format) & FOOTNOTE

Task 14:

TABLE: Rows, Columns, Split, Merge, Color, Delete, Add, Alignment Border Styles…
MAILMERGE: Letter Format, Creating Data base, Mail merge wizard.

SPREAD SHEET & PRESENTATION:

Task 15:

Working with Student Marks Data and generate graphs.
BASICS: Grid lines (add, del, merge, hide), Mouse Actions (select, drag, move), Auto fill, Color, Sheets (add, del, move, Rename), Date Formats
FUNCTIONS: SUM, AVERAGE & MEAN AVERAGE, STANDARD DEVIATION, IF & NESTEDIF/IFAND, COUNT & COUNTIF, RANK, MAX&MIN, MAX2 (LEVEL)

Task 16:

Working with Sort, Filter, and LOOKUP Features:
Hyperlink, Reference value, V/H LOOKUP.

Task 17:

Working with Presentations
Features to be covered: PPT orientation, Slide Layout, Master Layout (slide, template and notes), Types of views (basics, presentation, slide sorter, notes etc.), Inserting –Background, textures, Design Templates, Hidden slides, audio/video, Animations, Time settings.

Task 18:

Developing a Static Web Application using HTML 5 and CSS 3 features.

A College Management System wanted to maintain all the details about the faculty of the concerned department. Each faculty data should be available in a separated page. The home page of the web application should display the list of faculty using css3 grid and flexbox layout. Upon selecting the faculty, the details must be opened in a separate page. Use HTML 5 and CSS 3 features to format the faculty page.

Task 19:

Develop an e-commerce Web Application using HTML 5 and CSS 3 features.

Develop a responsive web application that adapts to various device widths. Design a breakpoint for mobile phone which should change the appearance of the navigation bar to toggle button that displays the all the navigation options. Use any e-commerce as a reference and include the necessary pages.

Task 20:

1. Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that computes the gross salary of an employee according to the following rules:
   i)If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.
4. ii)If the basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic salary is entered interactively through the keyboard.

Task 21:

1. Write a shell script that accepts two integers as its arguments and computers the value of the first number raised to the power of the second number.
2. Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
3. Write shell script that takes a login name as command – line argument and reports when that person logs in.

Note: Tasks 1 to 15 are mandatory, and Tasks 16 to 21 are optional.

Course Outcomes: At the end of the course, the student should be able to

CO 1: Navigate through the Linux file system and specify access permission on new or existing files.
CO 2: Express and implement pattern-matching techniques using grep.
CO 3: Develop shell scripts involving shell and regular variables, and list and command-oriented control structures.
CO 4: Design blogs and beautiful web pages using HML and CSS.
CO 5: Format word documents with various contents such as tables, and figures, extract data analytics from excel file and present the insights using PowerPoint.

References:

1. Responsive Web Design with HTML5 and CSS 3, Ben Frain, 3rd Edition, Packt Publication, 2012.
2. Unix for programmer and users, 3rd edition, Graham Glass, King Ables, Pearson Education, 2003.
3. Unix Programming environment, Kernighan and Pike, PHI/Pearson Education,1984.
4. Headfirst WordPress, Jeff Sairto, O’Reilly Media, Inc., First Edition, 2010.

Course Code: 22BS151/101

APPLIED PHYSICS

(Common to All Branches)

Instruction	:	3 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	1	Semester End Examination	:	60 Marks
Credits	:	4	Semester End Exam Duration	:	3 Hours

Course Objectives: The objectives of this course for the student are to:

1. Study the characteristics of lasers and optical fibres.
2. Understand the basic principles of quantum physics and the band theory of solids.
3. Understand the underlying mechanism involved in the construction and working principles of various semiconductor devices.
4. Study the fundamental concepts related to the dielectric, magnetic, and energy materials.
5. Identify the importance of nanoscale, quantum confinement, and various fabrications techniques.

Unit I – Laser and Fiber Optics

Laser: Characteristics of laser light, stimulated absorption, spontaneous and stimulated emission of radiation, evaluation of the relation between Einstein coefficients, population inversion, meta-stable state, laser components, Ruby laser, He-Ne laser, Applications of lasers.

Fiber optics: Structure of optical fiber, the principle of propagation of light through optical fiber, acceptance angle, numerical aperture, types of optical fibers: step index and graded index. Signal attenuation in optical fibers - attenuation coefficient, Bending losses. optical fiber communication and application of optical fibers.

Unit II – Quantum Physics and Solids

Quantum Physics: Introduction to quantum physics, blackbody radiation – Stefan-Boltzmann's law, Wein's and Rayleigh-Jean's law, Planck's radiation law - photoelectric effect–de-Broglie hypothesis -matter waves - Heisenberg uncertainty principle - Born interpretation of the wave function – time independent Schrodinger wave equation - particle in one dimensional potential box.

Solids: Free electron theory (Drude & Lorentz, Sommerfeld) - Fermi-Dirac distribution - Bloch's theorem -Kronig-Penney model – E-K diagram- effective mass of the electron (qualitative) –origin of energy bands- classification of solids.

Unit III – Semiconductors and Devices

Intrinsic and extrinsic semiconductors (qualitative) – Fermi level in a semiconductor and its variation with charge carrier concentration and temperature - Hall effect - direct and indirect band gap semiconductors - construction, principle of operation and characteristics of P-N Junction diode, Zener diode – LED, and solar cell, their structure, materials, working principle and characteristics.

Unit IV – Dielectric, Magnetic and Energy Materials

Dielectric Materials: Basic definitions- types of polarizations (qualitative) - ferroelectric, piezoelectric and pyroelectric materials – applications – liquid crystal displays (LCD).

Magnetic Materials: Classification of magnetic materials into dia, para, ferro, antiferro and ferrimagnetic materials - Hysteresis - soft and hard magnetic materials - magnetostriction – applications of magnetic materials.

Energy Materials: Conductivity of liquid and solid electrolytes – Superionic conductors – Materials and electrolytes for super capacitors.

Unit V – Nanotechnology

Nanoscale, quantum confinement, surface to volume ratio, bottom-up fabrication: sol-gel, precipitation, chemical vapor deposition (CVD) – top-down fabrication: ball milling - physical vapor deposition (PVD) - characterization techniques - XRD, SEM & TEM - applications of nanomaterials.

Course Outcomes: At the end of the course the student will be able to:

CO 1 : Understand various aspects of Lasers and Optical fiber and their applications in diverse fields.

CO 2 : Understand physical world from fundamental point of view by the concepts of Quantum mechanics and visualize the difference between conductor, semiconductor, and an insulator by classification of solids.

CO 3 : Identify the role of semiconductor devices in science and engineering Applications.

CO 4 : Explore the fundamental properties of dielectric, magnetic and energy materials, their applications.

CO 5 : Appreciate the features and applications of Nanomaterials.

Textbooks:

1. Applied physics, P. K. Palanisamy, Scitech Publications (India) Pvt Limited
2. Electronic Devices and Circuits, Milliman and Halkias, McGraw-Hill publications.
3. Essentials of Nanoscience & Nanotechnology, Narasimha Reddy Katta, 1st Edition, Nano Digest (Editor) 2021.

References:

1. Introduction to Solid State Physics, C. Kittel, 8th Edition, Wiley India.
2. Engineering physics, Hitendra K. Malik and A. K. Singh, McGraw-Hill publications.
3. Electronic devices and circuits, S. Salivahanan and N. Suresh Kumar, McGraw-Hill publications.

Course Code: 22BS154

APPLIED LINEAR ALGEBRA

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS and IT Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the concept of rank of a matrix and application of rank to determine the consistency of a linear system of equations.
2. To learn and evaluate eigen values, eigen vectors of a matrix and hence find the Modal matrix of the corresponding linear transformation that transforms to Spectral matrix
3. To understand the concepts of derivatives of matrices.
4. To learn about vector spaces and inner product spaces and appreciate in Gram-Schmidt Orthogonalization process.
5. To learn various decomposition methods.

Unit I – Matrices and Linear systems

Types of matrices (Real), Rank of a matrix by Echelon form, Inverse of square matrices by Gauss-Jordan method and non-square matrices by Moore-Penrose method, System of linear equations: Consistency-Solving system of Homogeneous and Non-Homogeneous equations by Gauss elimination method. Applications to traffic flow and electrical circuits.

Unit II - Eigen values, Eigen vectors and Quadratic forms

Projection and Rotation matrices. Eigen values and Eigen vectors, properties (without proofs), Diagonalization of a matrix. Quadratic forms and its nature. Reduction of a quadratic form to canonical form by orthogonal transformation.

Unit III - Matrix Calculus

Scalar and Vector functions- Derivatives of matrices and Vectors: Denominator and Numerator Layout, Derivative of scalar function w.r.t Vector (Gradient) and vice versa, Derivative of Vector w.r.t Vector (Jacobian). Derivative of scalar functions of a matrix w.r.t a vector. Chain Rule. Matrix Differentials.

Unit IV - Vector Spaces

Definition of a Vector space, Subspace, Linear combination of vectors, Linear Dependence/Independence of vectors, linear span, Basis and dimension.
Row space, Column space and Null space of a matrix- Inner product and Outer product of vectors, Norm of a vector, Orthogonal projection of vectors, Gram-Schmidt Orthogonalization.
Applications to Least squares approximation.

Unit V - Matrix Decompositions

LU factorization, Singular Value decomposition, QR decomposition and Cholesky decomposition

Course Outcomes: At the end of the course, the student should be able to

CO1	:	Model high dimensional data using matrices and evaluate the rank of matrices.
CO2	:	Evaluate Eigen values, Eigen vectors and find the Modal matrix under a linear transformation.
CO3	:	Apply the concept of matrix derivatives in various machine learning techniques.
CO4	:	Appreciate the concept of vector spaces and solve real world engineering problems using Least squares approximations.
CO5	:	Demonstrate the decomposition techniques of matrix to optimize the computational complexity.

Textbooks:

1. "Linear Algebra and Its Applications", David C. Lay, 4th Edition, Addison-Wesley, 2012.
2. "Applied Linear Algebra" Peter J. Olver. Chehrzad Shakiban, 2nd Edition, Springer International Publishing, 2018.

References:

1. "Mathematics for Machine Learning", Marc Peter Deisenroth, Cambridge University Press, 2020.
2. "Foundations of Data Science", Avrim Blum, Cambridge University Press, 2020.
3. "Introduction to Applied Linear Algebra", Stephen Boyd Cambridge University Press, 2018.
4. "Introduction to Linear Algebra", Gilbert Strang, 5th Edition, Wellesley-Cambridge Press, 2016.

Course Code: 22HS151/101

ENGLISH FOR SKILL ENHANCEMENT

(Common to All Branches)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives: Students will be able to

1. Use the English language appropriately according to context, culture, and domain.
2. Develop academic and professional writing competence.
3. Assess and interpret texts by reading, comprehending, and learning new vocabulary.
4. Recognize and analyze the features of a variety of genres.
5. Enhance language skills so that they can comprehend engineering subjects and hone their soft skills to deal with psychological and emotional challenges effectively.

Unit I

Chapter entitled **Toasted English by R.K. Narayan**

Vocabulary: The Concept of Word Formation -The Use of Prefixes and Suffixes -Acquaintance with Prefixes and Suffixes from Foreign Languages to form Derivatives - Synonyms and Antonyms

Grammar: Identifying Common Errors in Writing with Reference to Articles and Prepositions.

Reading: Reading and Its Importance- Techniques for Effective Reading.

Writing: Sentence Structures -Use of Phrases and Clauses in Sentences- Importance of Proper Punctuation- Techniques for Writing precisely – Paragraph Writing – Types, Structures and Features of a Paragraph - Creating Coherence-Organizing Principles of Paragraphs in Documents.

Unit II

Chapter entitled **Appro JRD by Sudha Murthy**

Vocabulary: Words Often Misspelt - Homophones, Homonyms and Homographs

Grammar: Identifying Common Errors in Writing with Reference to Noun-pronoun Agreement and Subject- verb Agreement.

Reading: Sub-Skills of Reading – Skimming and Scanning – Exercises for Practice

Writing: Nature and Style of Writing- Classifying- Providing Examples or Evidence.

Unit III

Chapter entitled **Lessons from Online Learning by F.Haider Alvi, Deborah Hurst et al**

Vocabulary: Words Often Confused - Words from Foreign Languages and their Use in English.

Grammar: Identifying Common Errors in Writing with Reference to Misplaced Modifiers and Tenses.

Reading: Sub-Skills of Reading – Intensive Reading and Extensive Reading
Exercises for Practice.

Writing: Format of a Formal Letter-Writing Formal Letters E.g.., Letter of Complaint, Letter of Requisition, Email Etiquette, Job Application with CV/Resume.

Unit IV

Chapter entitled **Art and Literature by Abdul Kalam**

Vocabulary: Standard Abbreviations in English

Grammar: Redundancies and Clichés in Oral and Written Communication.

Reading: Survey, Question, Read, Recite and Review (SQ3R Method) - Exercises for Practice

Writing: Writing Practices- Essay Writing-Writing Introduction and Conclusion -Précis Writing.

Unit V

Chapter entitled **Go, Kiss the World by Subroto Bagchi**

Vocabulary: Technical Vocabulary and their Usage

Grammar: Common Errors in English

Reading: Reading Comprehension-Exercises for Practice

Writing: Technical Reports- Introduction – Characteristics of a Report – Categories of Report Formats.

<u>Note</u>: Listening and Speaking Skills which are given under Unit-6 in AICTE Model Curriculum are covered in the syllabus of ELCS Lab Course.

Course Outcomes: At the end of the Semester, students will be able to:

CO 1 : Choose appropriate vocabulary and sentence structures for oral and written communication suitable to the context and culture.

CO 2 : Communicate effectively in various professional contexts through oral and written communication.

CO 3 : Comprehend, emphasize, conceptualize and evaluate the given texts and other authentic texts such as magazines, newspaper articles etc.

CO 4 : Understand explicit and implicit meaning and draw inference from the given text.

CO 5 : Evaluate their language skills and soft skills to handle personal and professional challenges.

Textbook:

1. "*English: Language, Context and Culture*" published by Orient Black Swan Pvt. Ltd, Hyderabad. 2022. Print.

References:

1. Liss, Davis. *Effective Academic Writing*, UK, OUP, 2000
2. Wood, F.T. *Remedial English Grammar,* India, Macmillan. 2007
3. Chaudhuri, Sinha. *Learn English: A Fun Book of Functional Language, Grammar and Vocabulary. (2nd ed.,).* Sage Publications India Pvt. Ltd. (2018)
4. Vishwamohan, Aysha. *English for Technical Communication for Engineering Students.* McGraw-Hill Education India Pvt. Ltd. (2013)
5. Swan, Michael. *Practical English Usage*. Oxford University Press. Fourth Edition. (2016)

Course Code: 22CS151

DATA STRUCTURES THROUGH ‘C’

(Common to All Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	3	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To introduce and impart knowledge to the student on the concepts of abstract data type, data structure, performance measurement, time and space complexities of algorithms.
2. To enable understanding of the student, towards a real-world problem-solving involving representation of data or physical entities in the program, processing them through a well-defined set of operations while giving persistence.
3. To enable the student, apply appropriate data structures to solve a complex problem.
4. To enable the student, analyze the solutions available for a problem, model, design and implement the best algorithm for an application development.

Unit I – Structures and Files

Structures and Unions: Defining a structure, accessing members of a structure, User-defined Data Types (typedef), Structures and Pointers, passing Structures to Functions, Self-referential Structures, Unions and Enumerated Data Types, Command Line Arguments.

Files: Opening and Closing a Data file, Creating a Data File, Processing a Data File, Unformatted Data Files.

Unit II - Linear Lists

Lists: Introduction to linear, non-linear data structures, What is a List, Operations on a List, List Implementation using Arrays and Linked Lists, Doubly Linked Lists.

Unit III – Stacks & Queues

Stacks: Stack ADT, Implementation of Stacks using Arrays and Linked lists. Applications of Stacks – infix to postfix, postfix evaluation of expressions, and their implementation

Queues: Queue ADT, Implementation of Queues using Arrays and Linked Lists, Implementation of Circular Queue using Arrays.

Unit IV – Trees

Introduction to Trees: Basic Tree concepts, Terminology, User Representation

Binary tree: Definition, Types of Binary Trees, Properties of Binary Trees, Binary Tree Traversals.

Binary Search Tree (BST): Definition, Operations: Traversals, insertion, deletion, Search, Binary Search Tree ADT implementation.

Unit V – Graphs

Graphs: Definition, Basic Concepts, Properties, types of graphs, Applications of graphs, Graph Storage Structures- Adjacency Matrix, Adjacency lists, Operations on Graphs: Insert Vertex, Delete Vertex, Add Edge, Delete Edge, Find Vertex Algorithms, Graph Traversals, Operations on Graphs Algorithms, Implementation of BFS, DFS.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Design and implement appropriate user-defined types to a given problem definition and apply various functions for processing files.
CO 2 : Understand basic concepts, Design and implement linear list data structures.
CO 3 : Implement stack and queue data structures and their application.
CO 4 : Assimilate the terminology of trees and implement binary tree operations in C.
CO 5 : Understand the representation of graph and traversal techniques.

Textbooks:

1. Data Structures: A Pseudocode Approach with C, Richard F. Gilberg, Behrouz A. Forouzan, 2nd Edition, Cengage Learning, 2004.
2. Fundamentals of Data structures in C, E. Horowitz, S. Sahni and Susan Anderson-Freed, 2nd Edition, Universities Press, 1993.

References:

1. Data Structures using C, R. Thareja, Oxford University Press, 2014
2. Data Structures, Schaum"s Outlines, S. Lipscutz, TMH, 2014
3. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, Career Monk, 2016

Course Code: 22IT151

PYTHON FOR COMPUTING

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS and IT Branches)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	2	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand decision control constructs, functions, and modules of Python Programming.
2. To acquire an in-depth understanding of data structures in Python for program design and development.
3. To learn file handling and advanced features of Python.

Unit I – Introduction to Python

Python Basics – Python Interpreter and IDLE environment, Basic Data Types, Variables, statements, expressions, Operators, Strings, Control Structures – Branching and looping structures, Simple programs.

Unit II – Functions and Modules

Introduction, Function Definition, Function Call, Variable Scope and Lifetime, the return statement, More on Defining Functions, Lambda Functions or Anonymous Functions, Documentation Strings, Good Programming Practices, Recursive Functions, Modules, Packages in Python, Standard Library modules, Globals (), Locals (), and Reload (), Function Redefinition, Functions as Objects.

Functional Programming – filter () Function, map () Function, reduce () Function.

Unit III – Built-in Data Structures

List - Sequence, *Lists* - Access Values in Lists, Updating Values in Lists, Nested Lists, Cloning Lists, Basic List Operations, List Methods.

Tuple - Creating Tuple, Utility of Tuples, Operations on Tuples, Nested Tuples, List Comprehension and Tuples, Advantages of Tuple over List.

Sets- Creating a Set, Set Operations.

Dictionaries - Creating a Dictionary, Basic Dictionary Operations, Nested Dictionaries, Built-in Dictionary Functions and Methods, Difference between a List and a Dictionary, String Formatting with Dictionaries.

Unit IV – File Handling and Advance Features of Python

Introduction, File Path, Types of Files, Opening and Closing Files, Reading and Writing Files, File Positions, Renaming and Deleting Files, Directory Methods.

Advance Features of Python: Python Exception Handling, Conditional Expressions, Comprehension Syntax, Packing and Unpacking of Sequences, Scopes and Namespaces, Modules, and the Import Statement.

Unit V – Object-Oriented Programming and Data Analysis with Python

Class Definitions, Object-Orientated concepts, Inheritance, and its types, Shallow and Deep Copying, and regular expressions.
Data Analysis with Python: Numpy - ndarray – Introduction, creating ndarray, data types for ndarray, operations between arrays and scalars, basic indexing, and slicing.

Course Outcomes: At the end of the course, the student should be able to

CO 1 : Write programs using various control structures of Python.
CO 2 : Use functions and modules for problem-solving.
CO 3 : Design and develop solutions to real-world problems using available data structures in Python Language.
CO 4 : Write programs using files & Advanced features of Python.
CO 5 : Know the importance of object-oriented programming in Python & analyze data with Numpy.

Textbooks:

1. Python Programming Using Problem Solving Approach, Reema Thareja, Oxford University Press 2017.
2. Python Programming: A Modular Approach, Sheetal Taneja and Naveen Kumar, Pearson, 2018.

References:

1. Programming in Python 3: A Complete Introduction to Python Language, Mark Summerfield, Second edition, Addison-Wesley, 2009.
2. Think Python: How to Think Like a Computer Scientist, Allen B. Downey, 2nd edition, Shroff / O‘Reilly Publishers, 2016.

Course Code: 22HS152/102

ENVIRONMENTAL SCIENCE

(Mandatory Course, Common to All Branches)

Instruction	:	3 Periods/ week	Sessional Marks	:	100
Tutorial	:	-			
Credits	:	-			

Course Objectives:

1. To create awareness on significance of ecosystems.
2. To emphasize the value of biodiversity and conservation of biodiversity.
3. To educate students about the importance of natural resources and their sustainable utilization.
4. To develop awareness in the students about the significance of environmental Pollution.
5. To create awareness regarding environmental management and to understand the environmental legislation.

Unit I - Ecosystems

Ecosystems: Definition of Ecosystem, Classification of ecosystem, Structure and Functions of Ecosystem, Pond and Grassland Ecosystems, Food Chains -Grazing and Detritus, Food web and Ecological Pyramids, Flow of Energy, Biogeochemical cycles: Carbon cycle and Nitrogen - Types of nitrogen fixation and cycle.

Unit II - Biodiversity

Biodiversity: Definition, Types of biodiversity (Species, Genetic and Ecosystem), Values of biodiversity- Ecological value and economic value, Hotspots of biodiversity-Western Ghats, Himalayas, Threats to biodiversity - Loss of habitat, Poaching, Invasion of species, Man-wildlife conflicts in Indian context. IUCN categories of bio diversity, Red data book-endangered species of India, Conservation of biodiversity: In-situ and Ex-situ conservation and wildlife conservation-Project tiger.

Unit III - Natural Resources

Classification of resources - Renewable and Non-renewable Resources Forest Resources–Uses of forests and over-exploitation of forests and causes of deforestation. Water Resources: Zones of a lake, environmental problems of a lake, Dams-Benefits and Problems, Mineral Resources: Classification of minerals, Methods of mining and Mining and its Environmental Impacts, Renewable Energy Resources/ Net zero concept: Solar Energy, Wind Energy Biomass energy.

Unit IV - Environmental Pollution

Air Pollution-Sources, classification, effects and control measures, Green-house gases-Causes and consequences of Global Warming, Kyoto Protocol, Ozone layer depletion, Montreal Protocol. Water Pollution-Sources, classification and effects, Wastewater Treatment Methods/Zero liquid discharge: Primary, secondary treatment, Effluent Treatment Plant (ETP), Brief account of Soil Pollution and Noise Pollution.

Unit V - EIA and Environmental Legislation

Definition and Scope of EIA, Base Line Data Acquisition, and Impact Assessment Methodologies-Check list method, Ad-hoc method, Leopold matrix method, EMP-advances in EMP Air(Prevention and Control of Pollution) Act-1981, Water(Prevention and Control of Pollution) Act-1974, Environment Protection Act-1986, Municipal Solid Waste- Classification and disposal methods and Biomedical Waste- Categories and disposal methods.

Field Trip: Study of ecosystems-Pond/ lake/ river/ forest, Visit to an urban/rural/industrial/ agricultural site, visit to STP/ ETP/ CETP/ Green building council.

Course Outcomes: At the end of the course, the student should be able to

CO1 : Define the concepts of Ecosystem.

CO2 : Explain and enunciate the value of biodiversity and its conservation.

CO3 : Discuss various natural resources and their importance, understand the advantages and disadvantages of renewable energy sources and technologies.

CO4 : Develop awareness on pollution control technologies and global atmospheric changes.

CO5 : Relate the importance of Environmental Impact Assessment and Environmental legislation in the management of the environment.

Textbooks:

1. Perspectives in Environmental Studies, Anubha Kaushik and C. P.Kaushik, 7th Edition, New Age International Pvt. Ltd. Publishers 2022.
2. Textbook of Environmental Science and Technology, M. Anji Reddy, B.S. Publications, 2013.

References:

1. Ecology and Environment, PD Sharma, 11th Edition, Rastogi publications,2011.
2. Essential Environmental Studies, SP. Misra and SN Pandey 3rd Edition, Ane Book Pvt. Ltd., 2012.

Course Code: 22BS181/131

APPLIED PHYSICS LAB

(Common to All Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives: The objectives of this course for the student to

1. Understand the optical phenomena such as diffraction, beam divergence of LASER beam, total internal reflection and bending losses in optical fiber.
2. Capable of handling instruments related to the Hall effect and photoelectric effect experiments and their measurements.
3. Understand the characteristics of various devices such as PN junction diode, Zener diode, BJT, LED and solar cell.
4. Able to measure the dielectric constant of a given material and study the behavior of B-H curve of ferromagnetic materials.
5. Recall the basic concepts of LCR circuit, RC circuit and dispersion of light through hands on experience and also understand the method of least squares fitting.

List of experiments: (Minimum 8 experiments are to be performed)

1. a) Determination of the beam divergence of the given LASER beam.
   b) Determination of the wavelength of the given LASER beam.
2. a) Determination of Acceptance Angle and Numerical Aperture of an optical fiber.
   b) Measurement of bending losses in optical fiber.
3. Determination of work function and Planck's constant using the photoelectric effect.
4. Determination of Hall co-efficient and carrier concentration of a given semiconductor.
5. V-I characteristics of a p-n junction diode and Zener diode.
6. Input and output characteristics of BJT (CE configuration).
7. a) V-I and L-I characteristics of the light-emitting diode (LED).
   b) V-I Characteristics of solar cell.
8. Determination of the Energy gap of a semiconductor.
9. Determination of the dielectric constant of a given material.
10. Study the B-H curve of a magnetic material.
11. Characteristics of series and parallel LCR circuits.
12. Measurement of the time constant of an RC circuit.
13. Understanding the method of least squares – torsional pendulum as an example.
14. Determination of dispersive power of material of prism using the spectrometer.

Course Outcomes: At the end of the course, the students will be able to:

CO 1 : Determine the wavelength of light by diffraction principle and learn methods to minimize the signal loss in optical fibers.

CO 2 : Understand the applications of the Photoelectric effect, develop skills to identify the type of semiconductors and determine charge carrier concentration in it using Hall effect.

CO 3 : Understand the applications of various semiconductor, and optoelectronic devices.

CO 4 : Gain knowledge of applications of dielectric materials and hysteresis behavior of magnetic materials.

CO 5 : Understands the concepts of resonance, charging and discharging of the capacitor, dispersion of light and carry out data analysis.

Textbooks:

1. "A Textbook of Practical Physics" S. Balasubramanian, M.N. Srinivasan - S Chand Publishers, 2017.

Course Code: 22ME181/131

ENGINEERING WORKSHOP

(Common to All Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To acquire skills in basic engineering practice.
2. To identify the hand tools and instruments.
3. To acquire measuring skills.
4. To acquire practical skills in the trades and understand safety practices.
5. To develop the right attitude and learn to work in a team at work place.

1.TRADES FOR EXERCISES: (Any four trades of the following)

i. Carpentry
ii. Fitting
iii. Tin smithy
iv. House wiring
v. Foundry

2. TRADES FOR DEMONSTRATION & EXPOSURE:

i. Machine Shop (Lathe operations)
ii. Power Tools
iii. Welding

Course Outcomes: At the end of the course, the student should be able to

CO 1	:	Acquire skills of basic engineering trades like Carpentry, Tin smithy etc.
CO 2	:	Demonstrate an understanding of and comply with workshop safety regulations.
CO 3	:	Identify and use marking out tools, hand tools, and measuring equipment and to work to prescribed tolerances.
CO 4	:	Apply the knowledge of the above trades in their day–to–day activities.
CO 5	:	Select appropriate equipment and consumables for required applications.

References:

1. Workshop Manual, P.Kannaiah & K.L.Narayana, 2nd Addition, Scitech Publishers, 2009.
2. Workshop Practice Manual, K. Venkat Reddy, 6th Addition, BS Publications, 2008.

Course Code: 22CS181

DATA STRUCTURES THROUGH 'C' LAB

(Common to All Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To develop skills to design and analyze simple linear and non-linear data structures and develop ADTs for stacks, queues, trees, and graphs to perform their corresponding operations.
2. To introduce the students to identifying and applying the suitable data structure for the given real-world problem.
3. To impart a practical understanding of how various information storage and retrieval techniques work.
4. To develop skills to Interpret syntax errors as reported by the compilers and to be able to identify and correct logical errors encountered at run time using debuggers like GDB.

All the programs need to be implemented in GDB mode in a Linux Environment

Task 1:

1. Write a C program to create a user-defined data-type Complex and implement addition, subtraction, and multiplication operations on complex numbers.
2. Create a user-defined data-type Student containing the fields roll no, name, and date of birth (by creating a user-defined type Date). Implement C functions to read the details of a student and create an array of students.

Task 2:

1. Write a C program to illustrate the user-defined data type union.
2. Write a C program to illustrate command-line arguments.

Task 3:

1. Write a C program to read the content of a given text file and count the number of characters, words, and lines in it (Read the file name as a command line argument).
2. Write a C program to read the content of a given text file, convert all lowercase letters into upper case and display it on the screen.
3. Write a C program to copy the contents of one file into another.

Task 4:

1. Write a C program to write the record list of Student types into a binary file student.dat. Re-open the file, read the records from the file, and display on the screen.

Task 5:

1. Write a C program to implement all the List operations using Arrays.

Task 6:

1. Write a C program to implement all the List operations using Linked Lists.

Task 7:

1. Write a C Program to implement StackADT using Arrays.

Task 8:

1. Write a C Program to implement StackADT using Linked Lists

Task 9:

1. Write a C Program to convert infix expression to postfix using stacks

Task 10:

1. Write a C Program to perform a postfix evaluation of an expression.

Task 11:

1. Write a C Program to implement QueueADT using Arrays.
2. Write a C Program to implement Circular Queue using Arrays.

Task 12:

1. Write a C Program to implement QueueADT using Linked Lists.

Task 13:

1. Write a C program to implement the following operations on Binary Search Tree: Insertion, deletion, and searching.
2. Write a C Program to perform traversals-preorder, in order and post order on a Binary Search Tree (BST).

Task 14:

1. Write a C Program to implement the Breadth First Traversal of a Graph.
2. Write a C Program to implement Depth First Traversal of a Graph.

Task 15:

1. Create a structure called Lab.

```
 struct Lab
{
char name [100];
float length.
float width.
float height.
int capacity.
};
```

Write a program to get the details of 'n' labs and to display the name, area and capacity of each, sorted by name in ascending order. Length, width and height of the building are given in feet. The area is to be computed with feet only.

Task 16:

1. Given a linked list of the form 1->2->3->4->5 swap two adjacent nodes, output of the example is 2->1->4->3->5
2. Given a linked list and value K, keep first K elements and remove next K elements, keep again K elements and remove next K elements.
3. Represent a polynomial as a linked list and write functions for polynomial addition.

Task 17:

1. C program to implement two stacks using a single array & check for overflow & underflow
2. C Program to Check String is Palindrome using Stack
3. C Program to Check if Expression is correctly Parenthesized

Task 18:

1. Implement a Queue using Stacks
2. Implement 2 queues in a single array, one from the front and the other from the rear of the array.

Task 19:

1. Write a program to process stock data. Use the internet on your local computer to gather data like stock code, stock name, the amount invested, etc. about at least 20 stocks. As each stock is read, insert it into a doubly linked list, and from there write the information to a file for persistent storage. Present a user-driven menu to select the action of his choice like insert, delete, display, search, etc.

Task 20:

1. Write a Program to perform customer billing after the purchase of products in a store. The bill should contain details like the Purchased Item code, Name, quantity purchased, price per unit and price for quantity purchased, and finally total bill to be paid. Develop the billing application for any store.

Note: Tasks 1 to 14 are mandatory, and Tasks 15 to 20 are optional.

Course Outcomes: At the end of the course, the student should be able to

CO 1: Implement file processing functions and be able to store, retrieve and process data in text and binary format

CO 2: Understand basic data structures such as arrays, and linked lists.

CO 3: Understand basic data structures such as stacks, queues, and circular Queues

CO 4: Implement operations on Binary Search Trees

CO 5: Solve problems involving graphs.

References:

1. C Programming and Data Structures, Behrouz A. Forouzan and Richard F. Gilberg, 3rd Edition, Cengage Learning, 2010.
2. Fundamentals of Data structures in C, E. Horowitz, S. Sahni and Susan Anderson-Freed, 2nd Edition, Universities Press, 1993.
3. Data Structures using C, R. Thareja, Oxford University Press, 2014.
4. Data Structures (Schaum's Outlines Series), S. Lipscutz, TMH, 2014.
5. Data Structures and Algorithms Made Easy, Narasimha Karumanchi, CareerMonk, 2016.

Course Code: 22IT181

PYTHON FOR COMPUTING LAB

(Common to CSE, CSE-AI&ML, CSE-CS, CSE-DS and IT Branches)

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1.5	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To train how to write, test, and debug simple Python programs.
2. To teach the usage of functions for structuring Python programs
3. To make student handle compound data using Python lists, tuples, dictionaries, etc.
4. To make students handle data in file processing.
5. To Write programs using object-oriented concepts in Python.

Exercises:

1. a. Write a Python program to create all possible strings by using 'a', 'e', 'i', 'o', 'u
   b. Write a Python program to create all possible permutations from a given collection of distinct numbers.
   c. Write a Python program to check the priority of the four operators (+, -, *, /).

2. a. Write a Python program that accepts a sequence of lines (blank line to terminate) as input and prints the lines as output (all characters in lower case).
   b. Write a Python program to check the validity of password input by users.
   - At least 1 number between [0-9].
   - At least 1 character from [$#@].
   - Minimum length 6 characters.
   - Maximum length 16 characters.
   - At least 1 letter between [a-z] and 1 letter between [A-Z].

3. a. Write a program to print the Floyd's triangle.
   b. Write a program to read month of the year as an integer. Then display the name of the month.
   c. Write a program that accepts any number and prints the number of digits in the number.

4. a. Write a Python function to check whether a number is in a given range.
   b. Write a Python function that prints out the first n rows of Pascal's triangle.
   c. Write a Python program to make a chain of function decorators (bold, italic, underline etc.) in Python.

5. a. Write a Python recursive program to calculate the sum of the positive integers of n+(n- 2) +(n-4)... (until n-x =< 0).
   b. Write a Python recursive program to calculate the harmonic sum of n-1
   c. Write a Python recursive program to find the greatest common divisor (gcd) of two integers.

6. a. Write a program that uses the lambda function to multiply two numbers.
   b. Write a program that passes a lambda function as an argument to another program to compute the cube of a number.
   c. Write a program to compute lambda(n) for all positive values of n where, lambda(n) can be recursively defined as lambda(n) = lambda(n/2) + 1 if n > 1

7. a. Write a Python program to find the list of words that are longer than n from a given list of words.
   b. Write a Python program to create a list by concatenating a given list whose range goes from 1 to n.
   c. Write a Python program to find missing and additional values in two lists.

8. a. Write a program to insert a value in a list at the specified location.
   b. Write a program to find the sum of all values in a list using reduce () function.

9. a. Write a Python program to remove empty tuple(s) from a list of tuples.
   b. Write a Python program to unzip a list of tuples into individual lists.
   c. Write a program that creates a list [‵a′, ‵b′, ‵c′] , then creates a tuple from that list. Now, do the opposite. That is, create the tuple (‵a′, ‵b′, ‵c′), and then create a list from it.

10. Write a program to make two sets of random integers and apply all set operations on them.

11. a. Write a Python program to sort a dictionary by key.
    b. Write a Python program to create and display all combinations of letters, selecting each letter from a different key in a dictionary.
    c. Write a Python program to create a dictionary from two lists without losing duplicate values.

12. a. Write a program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.

13. a. Write A Program that Reads a text file and counts the number of occurrences ofa given word.
    b. Write a program to compare two files.
    c. Write programs that exchange the contents of two files.

14. a. Write a program to count the number of records stored in the file employee.
    b. Write a program to merge two files into a third file. The names of the files must be entered using command line arguments.
    c. Write a function program to read the data from a file and count the total number of lines and words in the file.

15. a. Demonstrate the handling of standard exceptions in python programming.
    b. Demonstrate the handling of user-defined exceptions in python programming.

16. a. Write a Python program to select a random element from a list, set, dictionary (value) and a file from a directory. Use random.choice ().
    b. Write a Python program to check if a function is a user-defined function or not. Use types. FunctionType, types.LambdaType ().
    c. Write a Python program to construct a Decimal from a float and a Decimal from a string. Also, represent the Decimal value as a tuple. Use decimal. Decimal

17. a. Write A Program to Create a Class that Performs Basic Calculator Operations
    b. Write A Program to Create a Class in which One Method Accepts a String from the User and Another print it.

18. a. How to create an empty and a full NumPy array?
    b. Create a Numpy array filled with all zeros
    c. Create a Numpy array filled with all ones
    d. Check whether a Numpy array contains a specified row?

19. a. Matrix Multiplication in NumPy.
    b. Get the eigenvalues of a matrix.
    c. How to Calculate the determinant of a matrix using NumPy?
    d. How to inverse a matrix using NumPy.

20 a. Replace NumPy array elements that don’t satisfy the given condition
   b. Return the indices of elements where the given condition is satisfied
   c. Replace NaN values with the average of columns
   d. Replace negative value with zero in NumPy array

21. a. Write a Python program to slice ndarray with in the given range.
    b. Write a Python Program to create a surface plot and mesh plot using Matplot lib.

Note: Programs from 1 to 14 are mandatory and those from 15 to 21 are optional.

Course Outcomes: At the end of the course, the student should be able to

CO 1	:	Write programs using various control structures of Python.
CO 2	:	Use functions and modules for problem-solving.
CO 3	:	Design and develop solutions to real-world problems using available data structures in Python Language.
CO 4	:	Write programs using files & Advanced features of Python.
CO 5	:	Know the importance of object-oriented programming in Python & analyze data with NumPy.

Textbooks:

1. Python Programming Using Problem Solving Approach, Reema Thareja, Oxford University Press 2017.
2. Python Programming: A Modular Approach, Sheetal Taneja and Naveen Kumar, Pearson, 2018.

References:

1. Programming in Python 3: A Complete Introduction to Python Language, Mark Summerfield, Second edition, Addison-Wesley, 2009.
2. Think Python: How to Think Like a Computer Scientist, Allen B. Downey, 2nd edition, Shroff / O‛Reilly Publishers, 2016.