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.

Note: 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 of a 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.

ACADEMIC REGULATIONS,

COURSE STRUCTURE

&

SYLLABUS

R22 REGULATIONS

CHOICE BASED CREDIT SYSTEM (CBCS)

II B.Tech.

Computer Science and Engineering

Applicable to batches admitted in the First year
from 2022-23 onwards

CVR COLLEGE OF ENGINEERING

An UGC Autonomous Institution with NAAC Grade 'A'

(Approved by AICTE & Govt. of Telangana and
Affiliated to JNT University, Hyderabad)
Vastunagar, Mangalpalli (V), Ibrahimpatan (M),
Ranga Reddy Dist., Pin – 501 510

CVR COLLEGE OF ENGINEERING

VISION

- To be a state of the art institution of engineering in pursuit of excellence, in the service of society.

MISSION

- To excel in providing quality education at undergraduate and graduate levels.
- To encourage research and innovation.
- To provide infrastructure and facilities to meet the latest technological needs.
- To establish Centres of Excellence through active interaction with industry.
- To nurture students towards holistic development with human values and ethics.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

- Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Consultancy, and Technological services to society.

MISSION

1. To produce the best quality Computer Science & Engineering professionals by imparting quality training, hands-on experience and value education.
2. To strengthen links with industry through partnerships and collaborative developmental works.
3. To attain self-sustainability and overall development through Research, Consultancy, and Development activities.
4. To extend technical expertise to other technical institutions of the region and play a lead role in imparting technical education.
5. To inculcate work ethics and commitment in students for their future endeavors to serve society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

PEO 1 : Employability: Computer Science & Engineering graduates will acquire the capability to apply their knowledge and skills to solve various kinds of computational engineering problems.

PEO 2 : Professionalism: Graduates will inculcate a professional attitude, interdisciplinary approach, ethics, and ability to relate computer engineering issues with social awareness.

PEO 3 : Managerial skills: Graduates will possess managerial skills to face challenges in the profession by working harmoniously in a team with effective communication skills.

PEO 4 : Continuous learning: Graduates will continue to learn and adapt in a world of constantly evolving technologies and pursue research towards academic excellence.

PEO 5 : Adaptability: Graduates of Computer Science & Engineering will have soft skills to adapt to the diverse global environment.

PROGRAM OUTCOMES (POs):

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Successfully design and implement algorithmic paradigms by using efficient programming language constructs, formal tools, and frameworks.

PSO2: Develop scalable and reliable distributed applications and data analytics pipelines by employing industry-agnostic technologies and secure software engineering models.

PSO3: Adapt cloud computing ecosystems and machine learning algorithms to develop smart and sustainable solutions complying the ethics of society and eventually emerge as entrepreneurs.

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution)
Vastunagar, Mangalpalli, Ibrahimpatan–501510

ACADEMIC REGULATIONS–2022
B.Tech. PROGRAMMES

(Effective for the students admitted into I-year from the Academic Year 2022-23 onwards)

1.0 Under-Graduate Degree Programme in Engineering & Technology (B.Tech.: Undergraduate Programme (UGP) in Engineering & Technology (E&T))

CVR College of Engineering is an autonomous institution under the University Grants Commission, affiliated to Jawaharlal Nehru Technological University, Hyderabad. The College offers 4-Year (8- Semesters) **Bachelor of Technology** (B.Tech.) Degree Programme, under Academic Regulations-2022(R22) with **Choice Based Credit System (CBCS**) with effect from the Academic Year **2022-23** onwards, in the following Branches of Engineering:

Table-1

S. No.	Branch
1	Civil Engineering (CE)
2	Computer Science and Engineering (CSE)
3	Computer Science and Engineering – Artificial Intelligence and Machine Learning (CSE-AI&ML)
4	Computer Science and Engineering – Cyber Security (CSE-CS)
5	Computer Science and Engineering – Data Science (CSE-DS)
6	Electronics and Communication Engineering (ECE)
7	Electrical and Electronics Engineering (EEE)
8	Electronics and Instrumentation Engineering (EIE)
9	Information Technology (IT)
10	Mechanical Engineering (ME)

2.0 Eligibility for Admission

2.1 Category - A (70% of the sanctioned seats):

Admission to the UGP under Category–A are made by the Convener TS EAMCET based on the merit rank obtained by the qualifying candidate at an Entrance Test TS EAMCET conducted by Telangana State Government.

2.2 Category – B (30% of the sanctioned seats):

Admissions to the UGP under Category–B are made by the Management of the College and ratified by Telangana State Council of Higher Education (TSCHE) based on the merit rank of TS EAMCET / Marks in the Qualifying examination (Intermediate / Class XII) as prescribed in relevant G.Os. from time to time.

2.3 The medium of instruction for the entire UGP in Engineering & Technology will be in **ENGLISH** only.

3.0 B.Tech. Programme (UGP) Structure

3.1 A student after securing admission shall complete the B.Tech. programme in a minimum period of **four** academic years (8 semesters), and a maximum period of **eight** academic years (16 semesters) starting from the date of commencement of the first-year first semester, failing which the student shall forfeit seat in B.Tech. course. Each student shall secure 160 credits (with CGPA ≥ 5) required for the completion of the undergraduate program and award of the B.Tech. degree.

3.2 UGC/AICTE/JNTUH specified Definitions/Descriptions are adopted appropriately for various terms and abbreviations used in these Academic Regulations/ Norms, which are listed below.

3.2.1 Semester Scheme

Each UGP is of 4 Academic Years (8 Semesters), with the year being divided into two Semesters of 22 weeks (≥ 90 instructional days) each, each Semester having - 'Continuous Internal Evaluation (**CIE**)' and 'Semester End Examination (**SEE**)'. Choice Based Credit System (**CBCS**) and Credit Based Semester System (**CBSS**) as denoted by UGC, and Curriculum / Course Structure as suggested by the AICTE are followed.

3.2.2 Credit Courses

All subjects/ courses are to be registered by the student in a semester to earn credits which shall be assigned to each subject/ course in an L: T: P: C (lecture periods: tutorial periods: practical periods: credits) structure based on the following general pattern.

Table-2

1 Hour Lecture/Theory course per week (L)	1 credit
1 Hour Tutorial per week (T)	1 credit
2 Hour Practical/Laboratory course per week (P)	1 credit
3 Hours Practical/ Laboratory course per week (P)	1.5 credit

Courses like Environmental Science, Constitution of India, Intellectual Property Rights, and Gender Sensitization Lab are mandatory courses. These courses will not carry any credits.

3.2.3 Subject / Course Classification

All subjects/ courses offered for the undergraduate programme in E&T (B.Tech. degree programmes) are broadly classified as follows. The College has followed almost all the guidelines issued by JNTUH/AICTE/UGC.

Table-3

S. No.	***Broad Course Classification***	***Course Group/ Category***	***Course Description***
1	**Foundation Courses (FC)**	BS – Basic Sciences	Includes Mathematics, Physics and Chemistry subjects
2		ES - Engineering Sciences	Includes Fundamental Engineering Subjects
3		HS – Humanities and Social Sciences	Includes subjects related to Humanities, Social Sciences and Management
4	**Core Courses (CoC)**	PC – Professional Core Courses	Includes core subjects related to the parent discipline/ department/ branch of Engineering.

5	**Elective Courses (EC)**	PE – Professional Electives	Includes Elective subjects related to the Parent Discipline/ Department/ Branch of Engineering
6		OE – Open Electives	Elective subjects include inter-disciplinary subjects or subjects in an area outside the parent discipline/ department/ branch of Engineering.
7	**Core Courses**	Project Work	B.Tech. Project or UG Project or UG Major Project or Project Stage I & II
8		Industry Training/ Internship/ Industry Oriented Mini-project/ Mini-Project/ Skill Development Courses	Industry Training/ Internship/ Industry Oriented Mini-Project/ Mini-Project/ Skill Development Courses
9		Seminar	Seminar/ Colloquium based on core contents related to parent discipline/ department/ branch of Engineering.
10	Minor Courses	**-**	1 or 2 Credit Courses (subset of HS)
11	**Mandatory Courses (MC)**	**-**	Mandatory Courses (non-credit)

3.2.4 Induction Programme (Mandatory Course)

An Induction Programme is conducted as per the guidelines given by the AICTE at the beginning of the first semester of the first year, as presented in the Course Structure.

4.0 Course Registration

4.1 A ‘Faculty Advisor or Counselor’ shall be assigned to a group of 20 students, who will advise him about the UGP, its Course Structure, and Curriculum, and Choice/Option for Subjects/Courses, based on their competence, progress, pre-requisites, and interest.

4.2 Academic Section of the College invites filled ‘Registration Forms’ from students apriori (before the beginning of the Semester), through ‘ON-LINE SUBMISSIONS’, ensuring ‘DATE and TIME Stamping’. The ON-LINE Registration requests for any ‘CURRENT SEMESTER’ shall be completed BEFORE the commencement of SEEs (Semester End Examinations) of the ‘PRECEDING SEMESTER’.

4.3 A student can apply for ON-LINE Registration ONLY AFTER obtaining the ‘WRITTEN APPROVAL’ from the Faculty Advisor, which should be submitted to the College Academic Section through the Head of the Department. A copy of the same shall be retained by the Head of the Department, the Faculty Advisor, and the students.

4.4 A student may be permitted to register for all the subjects/ courses in a semester as specified in the course structure with the maximum additional subject (s)/course (s) limited to 6 Credits (any 2 elective subjects), based on **progress** and SGPA/ CGPA, and completion of the ‘**pre-requisites’** as indicated for various subjects/ courses, in the department course structure and syllabus contents.

4.5 Choice for ‘**additional subjects/ courses**, not more than any 2 elective subjects in any Semester, must be clearly indicated, which needs the specific approval and signature of the Faculty Advisor/Mentor/HOD.

4.6 If the student submits ambiguous choices or multiple options or erroneous entries during ON-LINE Registration for the Subject (s) /Course(s) under a given/ specified Course Group/ Category as listed in the Course Structure, only the first mentioned Subject / Course in that Category will be taken into consideration.

4.7 Subject / Course options exercised through ON-LINE Registration are final and CANNOT be changed, or interchanged; further, alternate choices will not be considered. However, if the Subject/ Course that has already been listed for Registration by the Head of the Department in a Semester could not be offered due to any unforeseen or unexpected reasons, then the student shall be allowed to have an alternate choice - either for a new Subject (subject to the offering of such a Subject) or for another existing Subject (subject to availability of seats), which may be considered. Such alternate arrangements will be made by the Head of the Department, with due notification and a time-framed schedule, within the **FIRST WEEK** from the commencement of class work for that Semester.

4.8 Dropping of Subjects / Courses may be permitted, ONLY AFTER obtaining prior approval from the Head of the Department (subject to retaining minimum Credits), 'within 15 Days of Time' from the beginning of the current Semester.

4.9 **Open Electives**: The students must choose two Open Electives (OE-I & II) from the list of Open Electives given by other departments. However, the student can opt for an Open Elective subject offered by his own (parent) department, if the student has not registered and not studied that subject under any category (Professional Core, Professional Electives, Mandatory Courses etc.) offered by parent department in any semester. Open Elective subjects already studied should not repeat/should not match with any category (Professional Core, Professional Electives, Mandatory Courses etc.) of subjects even in the forthcoming semesters.

4.10 **Professional Electives**: The students must choose **five** Professional Electives (PE-I to V) from the list of professional electives given.

5.0 Subjects/ courses to be offered

5.1 A subject/ course may be offered to the students, **only if** a minimum of 15 students opt for it.

5.2 More than **one faculty member** may offer the **same subject** (lab/ practical may be included with the corresponding theory subject in the same semester) in any semester. However, the selection of choice for students will be on a **first come first served** basis and CGPA criterion' (i.e., the first focus shall be on early **online entry** from the student for registration in that semester, and the second focus, if needed, will be on CGPA of the student).

5.3 If more entries for registration of a subject come into the picture, then the Head of the Department concerned shall decide, whether to offer such a subject/course for **two (or multiple) sections**.

5.4 In case of options coming from students of other departments/ branches/ disciplines (not considering **open electives**), first **priority** shall be given to the student of the '**parent department**'.

6.0 Attendance requirements:

6.1 A student shall be eligible to appear for the Semester End Examinations, if the student acquires a minimum of 75% of attendance in aggregate of all the subjects/ courses (including attendance in mandatory courses like Environmental Science, Constitution of India, Intellectual Property Rights, and Gender Sensitization Lab) for that semester. **Two periods** of attendance for each theory subject shall be considered if the student appears for the mid-term examination of that subject.

6.2 Shortage of attendance in aggregate up to 10% (65% and above, and below 75%) in each semester may be condoned by the college academic committee on genuine and valid grounds, based on the student's representation with supporting evidence.

6.3 A stipulated fee shall be payable for condoning of shortage of attendance.

6.4 **Shortage of attendance below 65% in aggregate shall in NO case be condoned.**

6.5 **Students whose shortage of attendance is not condoned in any semester are not eligible to take semester end examinations of that semester. They get detained and their registration for that semester shall stand canceled,** including all academic credentials (internal marks etc.,) of that semester**. They will not be promoted to the next semester.** They may seek re-registration for all those subjects registered in that semester in which the student is detained, by seeking re-admission into that semester as and when offered; if there are any professional electives and/ or open electives, the same may also be re-registered if offered. However, if those electives are not offered in later semesters, then alternate electives may be chosen from the **same** set of elective subjects offered under that category.

6.6 A student fulfilling the attendance requirement in the present semester shall not be eligible for readmission into the same class.

7.0 Academic Requirements

The following academic requirements must be satisfied, in addition to the attendance requirements mentioned in Item No. **6**.

7.1 A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (14 marks out of 40 marks) in the Continuous Internal Evaluation (CIE), not less than 35% (21 marks out of 60 marks) in the Semester End Examination, and a minimum of 40% (40 marks out of 100 marks) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together; in terms of letter grades, this implies securing **'P'** grade or above in that subject/ course.

7.2 A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to Real-time Research Project (or) Field Based Research Project (or) Industry Oriented Mini Project (or) Internship (or) Seminar if the student secures not less than 40% marks (i.e., 40 out of 100 allotted marks) in each of them. The student is deemed to have failed, if he (i) does not submit a report on Industry Oriented Mini Project/Internship, or (ii) does not make a presentation of the same before the evaluation committee as per schedule, or (iii) secures less than 40% marks in Real-time Research Project (or) Field Based Research Project (or) Industry Oriented Mini Project (or) Internship evaluations.

A student may reappear once for each of the above evaluations when they are scheduled again; if the student fails in such 'one reappearance' evaluation also, the student has to reappear for the same in the next subsequent semester, as and when it is scheduled.

7.3 Promotion Rules

Table-4

S. No.	Promotion	Conditions to be fulfilled
1	First-year first-semester to the first-year second-semester	Regular course of study of the first-year of the first- semester.
2	First-year second-semester to the second-year first-semester	(i) Regular course of study of first-year second-semester. (ii) Must have secured at least 20 credits out of 40 credits i.e., 50% credits up to the first-year second-semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or not.
3.	Second-year first-semester to the Second-year second-semester	Regular course of study of second-year first-semester

4	Second-year second-semester to Third-year first- semester	(i) Regular course of study of second-year second-semester. (ii) Must have secured at least 48 credits out of 80 credits i.e., 60% credits up to second-year second-semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or not.
5	Third-year first-semester to Third-year second-semester	Regular course of study of Third-year first-semester.
6	Third-year second-semester to Fourth-year first-semester	(i) Regular course of study of Third-year second-semester. (ii) Must have secured at least 72 credits out of 120 credits i.e., 60% credits up to Third-year second-semester from all the relevant regular and supplementary examinations, whether the student takes those examinations or not.
7	Fourth-year first-semester to Fourth-year second-semester	Regular course of study of Fourth-year first-semester.

7.4 A student (i) shall register for all courses/subjects covering 160 credits as specified and listed in the course structure, (ii) fulfill all the attendance and academic requirements for 160 credits, (iii) earn all 160 credits by securing SGPA ≥ 5.0 (in each semester), and CGPA ≥ 5 (at the end of 8 semesters), (iv) **passes all the mandatory courses,** to successfully complete the undergraduate programme. The performance of the student in these 160 credits shall be considered for the calculation of the final CGPA **(at the end of the undergraduate programme) and** shall be indicated in the grade card/marks memo of the IV-year II semester.

7.5 If a student registers for '**extra subjects'** (in the parent department or other departments/branches of Engineering) other than those listed subjects totaling 160 credits as specified in the course structure of his department, the performances in those '**extra subjects**' (although evaluated and graded using the same procedure as that of the required 160 credits) will not be considered while calculating the SGPA and CGPA. For such '**extra subjects'** registered, the percentage of marks and letter grade alone will be indicated in the grade card/marks memo as a performance measure, subject to completion of the attendance and academic requirements as stated in regulations items 6 and 7.1 – 7.4 above.

7.6 A student eligible to appear in the Semester End Examination for any subject/course but absent from it or failed (thereby failing to secure **'P'** grade or above) may reappear for that subject/ course in the supplementary examination as and when conducted. In such cases, internal marks (CIE) assessed earlier for that subject/ course will be carried over and added to the marks to be obtained in the SEE supplementary examinations for evaluating performance in that subject.

7.7 A student **detained in a semester due to a shortage of attendance may be re-admitted in the same semester in the next academic year for the fulfillment of academic requirements**. The academic regulations under which a student has been re-admitted shall be applicable. Further, no grade allotments or SGPA/ CGPA calculations will be done for the entire semester in which the student has been detained.

7.8 A student detained **due to lack of credits, shall be promoted to the next academic year only after acquiring the required number of academic credits.** The academic regulations under which the student has been readmitted shall be applicable to him.

8.0 Evaluation – Distribution, and Weightage of Marks

8.1 The performance of a student in every subject/course (including practicals and Project Stage – I & II) will be evaluated for 100 marks each, with 40 marks allotted for CIE (Continuous Internal Evaluation) and 60 marks for SEE (Semester End-Examination).

8.2 In CIE, for theory subjects, during a semester, there shall be two mid-term examinations. Each Mid-Term examination consists of two parts i) **Part – A** for 10 marks, ii) **Part – B** for 20 marks with a total duration of 2 hours as follows:

1. Mid Term Examination for 30 marks:
   a. Part - A: Objective/quiz paper for 10 marks.
   b. Part - B: Descriptive paper for 20 marks.

The objective/quiz paper is set with multiple choice, fill-in the blanks and match the following type of questions for a total of 10 marks. The descriptive paper shall contain 6 full questions out of which, the student must answer 4 questions, each carrying 5 marks. The **average of the two Mid Term Examinations** shall be taken as the final marks for Mid Term Examination (for 30 marks).

The remaining 10 marks of Continuous Internal Evaluation are distributed as:

2. Assignment for 5 marks. (**Average of 2 Assignments** each for 5 marks)
3. Subject Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the concerned subject for 5 marks.

While the first mid-term examination shall be conducted on 50% of the syllabus, the second mid-term examination shall be conducted on the remaining 50% of the syllabus.

Five (5) marks are allocated for assignments (as specified by the subject teacher concerned). The first assignment should be submitted before the conduct of the first mid-term examination, and the second assignment should be submitted before the conduct of the second mid-term examination. The average of the two assignments shall be taken as the final marks for assignment (for 5 marks).

Subject Viva-Voce/PPT/Poster Presentation/ Case Study on a topic in the subject concerned for 5 marks before II Mid-Term Examination.

The student, in each subject, shall have to earn 35% of marks (i.e., 14 marks out of 40 marks) in CIE, 35% of marks (i.e., 21 marks out of 60) in SEE, and Overall, 40% of marks (i.e., 40 marks out of 100 marks) both CIE and SEE marks put together.

The student is eligible to write the Semester End Examination of the concerned subject if the student scores ≥ 35% (14 marks) of 40 Continuous Internal Examination (CIE) marks.

In case, the student appears for the Semester End Examination (SEE) of the concerned subject but not scored a minimum 35 % of CIE marks (14 marks out of 40 internal marks), his/her performance in that subject in SEE shall stand canceled inspite of appearing the SEE.

There is NO Computer Based Test (CBT) for R22 regulations.

The details of the Semester End Examination question paper pattern are as follows:

8.2.1 The Semester End Examinations (SEE), for theory subjects, will be conducted for 60 marks consisting of two parts viz. i) **Part- A** for 10 marks, ii) **Part - B** for 50 marks.

- Part-A is compulsory that consists of ten sub-questions from all units carrying equal marks.
- Part-B consists of five questions (numbered from 2 to 6) carrying 10 marks each. Each of these questions is from a unit and may contain sub-questions. For each question there will be an "either" "or" choice, which means that there will be two questions from each unit and the student should answer either of the two questions.
- The duration of the Semester End Examination is 3 hours.

8.2.2 For the subject, Computer Aided Engineering Graphics, the Continuous Internal Evaluation (CIE) and Semester End Examinations (SEE) evaluation patterns are same as for other theory subjects.

8.3 For practical subjects there shall be a Continuous Internal Evaluation (CIE) during the semester for 40 marks and 60 marks for the Semester End Examination. Out of the 40 marks for internal evaluation:

1. A write-up on the day-to-day experiment in the laboratory (in terms of aim, components/procedure, expected outcome) which shall be evaluated for 10 marks.
2. **10 marks for viva-voce (**or) tutorial (or) case study (or) application (or) poster presentation of the course concerned.
3. Internal practical examination conducted by the laboratory teacher concerned shall be evaluated for 10 marks.
4. The remaining 10 marks are for the Laboratory Report/Project and Presentation, which consists of the Design (or) Software /Hardware Model Presentation (or) App Development (or) Prototype Presentation submission which shall be evaluated after completion of the laboratory course and before the semester end practical examination.

The Semester End Examination shall be conducted with an external examiner and the laboratory teacher. The external examiner shall be appointed from the cluster / other colleges which will be decided by the examination branch of the University.

In the Semester End Examination held for 3 hours, total 60 marks are divided and allocated as shown below:

1. 10 marks for write-up
2. 15 for experiment/program
3. 15 for evaluation of results
4. 10 marks for presentation on another experiment/program in the same laboratory course and
5. 10 marks for viva-voce on the concerned laboratory course.

- The student, in each subject, shall have to earn 35% of marks (i.e., 14 marks out of 40 marks) in CIE, 35% of marks (i.e., 21 marks out of 60) in SEE and Overall 40% of marks (i.e., 40 marks out of 100 marks) both CIE and SEE marks put together.

The student is eligible to write the Semester End Examination of the concerned subject if the student scores ≥ 35% (14 marks) of 40 Continuous Internal Examination (CIE) marks.

In case the student appears for the Semester End Examination (SEE) of the concerned subject but has not scored a minimum of 35% of CIE marks (14 marks out of 40 internal marks), his performance in that subject in SEE shall stand cancelled inspite of appearing for the SEE.

8.4 The evaluation of courses having ONLY internal marks in II Year II Semester is as follows:

II Year II Semester *Real-Time (or) Field-based Research Project* course: The internal evaluation is for 50 marks, and it shall take place during the I Mid-Term examinations and II Mid-Term examinations. The average marks of two Mid-Term examinations are the final 50 marks. The student shall have to earn 40%, i.e., 20 marks out of 50 marks from an average of the two examinations. There shall be NO external evaluation. The student is deemed to have failed if he (i) does not submit a report on the Project, or (ii) does not make a presentation of the same before the internal committee as per schedule, or (iii) secures less than 40% marks in this course.

8.5 There shall be an Industry training (or) Internship (or) Industry oriented Mini-project (or) Skill Development Courses (or) Paper presentation in a reputed journal (or) Industry Oriented Mini Project in collaboration with an industry of their specialization. Students shall register for this immediately after II-Year II Semester Examinations and pursue it during summer vacation/semester break & during III Year without affecting regular course work. Internship at a reputed organization (or) Skill development courses (or) Paper presentation in a reputed journal (or) Industry Oriented Mini Project shall be submitted in a report form and presented before the committee in the III-year II semester before Semester End Examination. It shall be evaluated for 100 external marks. The committee consists of an External Examiner, the Head of the Department, a Supervisor of the Industry Oriented Mini Project (or) Internship, etc., an Internal Supervisor, and a Senior Faculty Member of the Department. There shall be **NO internal marks** for Industry Training (or) Internship (or) Mini-Project (or) Skill Development Courses (or) Paper Presentation in a reputed journal (or) Industry Oriented Mini Project.

8.6 The UG project shall be initiated at the end of the IV Year I Semester and the duration of the project work is one semester. The student must present Project Stage – I during the IV Year I Semester before II Mid examinations, in consultation with his supervisor, the title, objective, and plan of action of his Project work to the departmental committee for approval before the commencement of the IV Year II Semester. Only after obtaining the approval of the departmental committee, the student can start his project work.

8.7 UG project work shall be carried out in two stages: Project Stage – I for approval of the project before Mid-II examinations in the IV Year I Semester and Project Stage – II during the IV Year II Semester. Students must submit a project work report at the end of the IV Year II Semester. The project shall be evaluated for 100 marks before the commencement of the SEE Theory examinations.

8.8 For Project Stage – I, the Departmental Committee consisting of the Head of the Department, the project supervisor, and a senior faculty member shall approve the project work to begin before the II Mid-Term examinations of the IV Year I Semester. The student is deemed to be not eligible to register for the Project work if he does not submit a report on Project Stage - I or does not make a presentation of the same before the evaluation committee as per schedule.

A student who has failed may reappear once for the above evaluation when it is scheduled again; if he fails in such 'one reappearance' evaluation also, he must reappear for the same in the next subsequent semester, as and when it is scheduled.

8.9 For Project Stage–II, the external examiner shall evaluate the project work for 60 marks and the internal project committee shall evaluate it for 40 marks. Out of 40 internal marks, the departmental committee consisting of the Head of the Department, the Project Supervisor and a Senior Faculty Member shall evaluate the project work for 20 marks and the Project Supervisor shall evaluate for 20 marks. The topics for Industry Oriented Mini Project/ Internship/SDC etc. and the main Project shall be different from the topic already taken. The student is deemed to have failed if he (i) does not submit a report on the Project, or (ii) does not make a presentation of the same before the External Examiner as per schedule, or (iii) secures less than 40% marks in the sum total of the CIE and SEE taken together.

For conducting viva-voce of the project, the Dean-Academics selects an external examiner from the list of experts in the relevant branch submitted by the Head of the Department.

A student who has failed may reappear once for the above evaluation when it is scheduled again; if a student fails in such 'one reappearance' evaluation also, he must reappear for the same in the next subsequent semester, as and when it is scheduled.

8.10 A student shall be given only a one-time chance to re-register for a maximum of two subjects in a semester:

If the internal marks secured by a student in the Continuous Internal Evaluation marks for 40 (Sum of average of two mid-term examinations consisting of Objective & descriptive parts, Average of two Assignments & Subject Viva- voce/PPT/ Poster presentation/ Case Study on a topic in the concerned subject) are less than 35% and failed in those subjects.

A student must re-register for the failed subject(s) for 40 marks within four weeks of the commencement of the classwork in the next academic year.

In the event of the student taking this chance, his Continuous Internal Evaluation marks for 40 and Semester End Examination marks for 60 obtained in the previous attempt stand cancelled.

8.11 For mandatory courses of Environmental Science, Constitution of India, Intellectual Property Rights, and Gender Sensitization lab, a student has to secure 40 marks out of 100 marks (i.e. 40% of the 100 marks allotted) in the Continuous Internal Evaluation for passing the subject/course. These marks should also be submitted along with the internal marks of other subjects.

8.12 No marks or letter grades shall be allotted for mandatory/non-credit courses. Only Pass/Fail shall be indicated in Grade Card.

9.0 Grading Procedure

9.1 Grades will be awarded to indicate the performance of students in each Theory Subject, Laboratory/Practicals/ Industry-Oriented Mini Project/Internship/SDC, and Project Stage. Based on the percentage of marks obtained (Continuous Internal Evaluation plus Semester End Examination, both taken together) as specified in item 8 above, a corresponding letter grade shall be given.

9.2 As a measure of the performance of a student, a 10-point absolute grading system using the following letter grades (as per UGC/AICTE guidelines) and corresponding percentage of marks shall be followed:

Table-5

% of Marks Secured in a Subject/Course (Class Intervals)	**Letter Grade (UGC Guidelines)**	**Grade Points**
Greater than or equal to 90%	S (Outstanding)	10
80 and less than 90%	A+ (Excellent)	9
70 and less than 80%	A (Very Good)	8
60 and less than 70%	B+ (Good)	7
50 and less than 60%	B (Average)	6
40 and less than 50%	P (Pass)	5
Below 40%	F (FAIL)	0
Absent	AB	0

9.3 A student who has obtained an '**F**' grade in any subject shall be deemed to have '**failed**' and is required to reappear as a 'supplementary student' in the Semester End Examination, as and when offered. In such cases, internal marks in those subjects will remain the same as those obtained earlier.

9.4 To a student who has not appeared for an examination in any subject, '**AB**' grade will be allocated in that subject, and he is deemed to have '**Failed**'. A student will be required to reappear as a 'supplementary student' in the Semester End Examination, as and when offered next. In this case also, the internal marks in those subjects will remain the same as those obtained earlier.

9.5 A letter grade does not indicate any specific percentage of marks secured by the student, but it indicates only the range of percentage of marks.

9.6 A student earns Grade Point (GP) in each subject/ course, based on the letter grade secured in that subject/ course. The corresponding Credit Points (CP) are computed by multiplying the grade point with credits for that particular subject/ course.

Credit Points (CP) = Grade Point (GP) x Credits .... For a course

9.7 A student passes the subject/ course only when **GP ≥ 5 ('P' grade or above)**

9.8 The Semester Grade Point Average (SGPA) is calculated by dividing the sum of Credit Points (ΣCP) secured from all subjects/ courses registered in a semester, by the total number of credits registered during that semester. SGPA is rounded off to **two** decimal places. SGPA is thus computed as

$$SGPA = \frac{(\sum_{i=1}^{N} C_i G_i)}{(\sum_{i=1}^{N} C_i)} \text{ ... For each semester}$$

where 'i' is the subject indicator index (considering all subjects in a semester), 'N' is the number of subjects '**registered**' for the semester (as specifically required and listed under the Course Structure of the parent department), C_i is the no. of credits allotted to the ith subject, and G_i represents the Grade Points (GP) corresponding to the letter grade awarded for that ith subject.

9.9 The Cumulative Grade Point Average (CGPA) is a measure of the overall cumulative performance of a student in all semesters considered for registration. The CGPA is the ratio of the total Credit Points secured by a student in **all** registered courses (of 160) in **all** semesters, and the total number of credits registered in **all** the semesters. CGPA is rounded off to **two** decimal places. CGPA is thus computed from the I year II semester onwards at the end of each semester as per the formula

$$CGPA = \frac{\left(\sum_{j=1}^{M} C_j G_j\right)}{\left(\sum_{j=1}^{M} C_j\right)}$$ ……… For all N semesters registered

(i.e., up to and inclusive of N semesters, N ≥ 2),

where '**M'** is the **total** no. of subjects (as specifically required and listed under the Course Structure of the parent department) the student has '**registered'** i.e., from the 1st semester onwards up to and inclusive of the 8th semester, 'j' is the subject indicator index (takes into account all subjects from 1 to 8 semesters), C_j is the number of credits allotted to the jth subject, and G_j represents the Grade Points (GP) corresponding to the letter grade awarded for the jth subject. After registration and completion of I year I semester, the SGPA of that semester itself may be taken as the CGPA, as there are no cumulative calculations.

Illustration of calculation of SGPA:

Course/Subject	Credits	Letter Grade	Grade Points	Credit Points
Course 1	4	A	8	4 x 8 = 32
Course 2	4	S	10	4 x 10 = 40
Course 3	4	P	5	4 x 5 = 20
Course 4	3	B	6	3 x 6 = 18
Course 5	3	A+	9	3 x 9 = 27
Course 6	3	P	5	3 x 5 = 15
	21			152

SGPA = 152/21 = 7.24

Illustration of Calculation of CGPA up to 3rd Semester:

Semester	Course/ Subject Title	Credits Allotted	Letter Grade Secured	Corresponding Grade Point (GP)	Credit Points (CP)
I	Course 1	3	A	8	24
I	Course 2	3	S	10	30
I	Course 3	3	B	6	18
I	Course 4	4	A	8	32
I	Course 5	3	A+	9	27
I	Course 6	4	P	5	20
II	Course 7	4	B	6	24
II	Course 8	4	A	8	32
II	Course 9	3	P	5	15
II	Course 10	3	S	10	30

II	Course 11	3	B+	7	21
II	Course 12	4	B	6	24
II	Course 13	4	A	8	32
II	Course 14	3	S	10	30
III	Course 15	2	A	8	16
III	Course 16	1	P	5	5
III	Course 17	4	S	10	40
III	Course 18	3	B+	7	21
III	Course 19	4	B	6	24
III	Course 20	4	A	8	32
III	Course 21	3	B+	7	21
	Total Credits	**69**		**Total Credit Points**	**518**

CGPA = 518/69 = 7.51

The calculation process of CGPA illustrated above will be followed for each subsequent semester until 8th semester. The CGPA obtained at the end of 8th semester will become the final CGPA secured for entire B.Tech. programme.

9.10 For merit ranking or comparison purposes or any other listing, **only** the '**rounded off'** values of the CGPAs will be used.

9.11 SGPA and CGPA of a semester will be mentioned in the semester Memorandum of Grades if all subjects of that semester are passed in first attempt. Otherwise, the SGPA and CGPA shall be mentioned only on the Memorandum of Grades in which sitting he passed his last exam of subjects of that semester. However, mandatory courses will not be taken into consideration.

10.0 Passing Standards

10.1 A student shall be declared successful or 'passed' in a semester, if he secures a GP ≥ 5 (**'P'** grade or above) in every subject/course in that semester (i.e. when the student gets an SGPA ≥ 5.0 at the end of that particular semester); and he shall be declared successful or 'passed' in the entire undergraduate programme, only when he gets a CGPA ≥5.00 (**'P'** grade or above) for the award of the degree as required.

10.2 After the completion of each semester, a grade card or grade sheet shall be issued to all the registered students of that semester, indicating the letter grades and credits earned. It will show the details of the courses registered (course code, title, number of credits, grade earned, etc.) and credits earned. **There is NO exemption of credits in any case.**

11.0 Declaration of results

11.1 Computation of SGPA and CGPA are done using the procedure listed from 9.6 to 9.9.

11.2 For final percentage of marks equivalent to the computed final CGPA, the following formula may be used.

% of Marks = (Final CGPA – 0.5) x 10

12.0 Award of Degree

12.1 A student who registers for all the specified subjects/ courses as listed in the Course Structure and secures the required number of 160 credits (with CGPA ≥5.0), within 8 academic years from the date of commencement of the first academic year, shall be declared to have '**qualified'** for the award of B.Tech. degree in the branch of Engineering selected at the time of admission.

12.2 A student who qualifies for the award of the degree as listed in item 12.1 shall be placed in the following classes:

12.2.1 A student with final CGPA (at the end of the undergraduate programme) > 8.00 and fulfilling the following conditions - shall be placed in '**First Class with Distinction'**. However, he

(i) Should have passed all the subjects/courses in '**First Appearance'** within the first 4 academic years (or 8 sequential semesters) from the date of commencement of first year first semester.

(ii) Should not have been detained or prevented from writing the Semester End Examinations in any semester due to shortage of attendance or any other reason.

A student not fulfilling any of the above conditions with final CGPA > 8 shall be placed in **'First Class'.**

12.2.2 Students with final CGPA (at the end of the undergraduate programme) ≥ 7.0 but <8.00 shall be placed in **'First Class'.**

12.2.3 Students with final CGPA (at the end of the undergraduate programme) ≥ 6.00 but <7.00, shall be placed in '**Second Class'**.

12.2.4 All other students who qualify for the award of the degree (as per item 12.1), with final CGPA (at the end of the undergraduate programme) ≥ 5.00 but < 6, shall be placed in '**Pass Class**'.

12.2.5 A student with final CGPA (at the end of the undergraduate programme) < 5.00 will not be eligible for award of degree.

12.3 Students fulfilling the conditions listed under item 12.2.1 alone will be eligible for award of '**Gold Medal**'.

12.4 Award of 2-Year B.Tech. Diploma Certificate

1. A student is awarded 2-Year UG Diploma Certificate in the concerned engineering branch on completion of all the academic requirements and earning all 80 credits (within 4 years from the date of admission) up to B. Tech. II Year II Semester if the student wants to exit the 4-Year B. Tech. programmme and requests for the 2-year B.Tech (UG) Diploma Certificate.
2. Once a student opts **for and is awarded 2-Year UG Diploma Certificate, the student will be permitted to join** in B. Tech. III Year I Semester and continue for completion of remaining years of study for 4-Year B. Tech. Degree ONLY in the next academic year along with the next batch of students. *However, if any student wishes to continue study after opting for exit, he/she should register for the subjects/courses in III Year I Semester before the commencement of classwork for that semester.*
3. *The students, who exit the 4-Year B. Tech. program after II Year of study and wish to re-join the B.Tech. programmme, must submit the 2-Year* B. Tech. (*UG) Diploma Certificate awarded to him, subject to the eligibility for completion of Course/Degree.*
4. A student may be permitted to take one year break after completion of II Year II Semester or B. Tech. III Year II Semester (with permission from the Principal of the college well in advance) and can re-enter the course in the **next Academic Year in the same college** and complete the course on fulfilling all the academic credentials within a stipulated duration i.e., double the duration of the course (Ex. within 8 Years for the 4-Year program).

13.0 Withholding of results

13.1 If the student has not paid the fees to the College at any stage or has dues pending due to any reason whatsoever, or if any case of indiscipline is pending, the result of the student may be withheld, and the student will not be allowed to go into the next higher semester. The award or issue of the degree may also be withheld in such cases.

14.0 Transitory Regulations

A. For students detained due to shortage of attendance:

1. A Student who has been detained in the I year of R18 Regulations due to lack of attendance shall be permitted to join I year I Semester of R22 Regulations and he is required to complete the study of B.Tech. programme within the stipulated period of eight academic years from the date of first admission in I Year.

2. A student who has been detained in any semester of II, III, and IV years of R18 Regulations for want of attendance, shall be permitted to join the corresponding semester of R22 Regulations and is required to complete the study of B.Tech. within the stipulated period of eight academic years from the date of the first admission in I Year. The R22 Academic Regulations under which a student has been readmitted shall be applicable to that student from that semester. See rule (C) for further Transitory Regulations.

B. For students detained due to shortage of credits:

3. A student of R18 Regulations who has been detained due to lack of credits, shall be promoted to the next semester of R22 Regulations only after acquiring the required number of credits as per the corresponding regulations of his/her first admission. The total credits required are 160 including both R18 & R22 Regulations. The student is required to complete the study of B.Tech. within the stipulated period of eight academic years from the year of first admission. The R22 Academic Regulations are applicable to a student from the year of readmission. See rule (C) for further Transitory Regulations.

C. For readmitted students in R22 Regulations:

4. A student who has failed in any subject under any regulation has to pass those subjects under the same regulations.

5. The maximum number of credits that a student acquires for the award of a degree, shall be the sum of the total number of credits secured in all the regulations of his/her study including R22 Regulations. **There is NO exemption of credits in any case**.

6. If a student is readmitted to R22 Regulations and has any subject with 80% of the syllabus common with his/her previous regulations, that particular subject in R22 Regulations will be substituted by another subject to be suggested by the College.

Note: If a student readmitted to R22 Regulations has not studied any subjects/topics in his/her earlier regulations of study which is a prerequisite for further subjects in R22 Regulations, the Heads of Departments concerned shall conduct remedial classes to cover those subjects/topics for the benefit of the students.

15.0 Student Transfers

(a) There shall be no branch transfers after completion of the admission process.

(b) A student seeking transfer to CVR College of Engineering from other Institutions affiliated to the JNTUH, after obtaining necessary permission from the State Government/ University must pass all the subjects at the previous institution.

(c) In case the student has failed in any subject, he has to take equivalent subject offered by this college and get a Pass grade. He should also obtain a Pass grade in those subjects of this college which the student has not studied at the previous institution, up to that semester when transfer was effective.

(d) For such of those transferred students with backlogs, the college will provide one chance to write the internal examinations in the failed subject and/or subject not studied in the curriculum of this college.

(e) Further, though the students have passed some of the subjects at the earlier institutions, if the same subjects are prescribed in different semesters of CVR College of Engineering, the students must study those subjects in spite of the fact that those subjects are repeated.

(f) Equivalent subjects will be notified by the college if required, on case-to-case basis as received from the University/as decided by the college. However, in case of Professional Electives and Open Electives, student has to opt for a subject among the subjects listed under each of the electives, as the case may be.

(g) For the completed semesters which the student studied previously at another institution/under a different scheme, Grade Points will be awarded as per the College rules and CGPA calculated after clearing backlogs, if any.

16.0 Scope

16.1 The academic regulations should be read as a whole, for the purpose of any interpretation.

16.2 In case of any doubt or ambiguity in interpretation of the above rules, the decision of the Vice Chancellor/Principal is final.

16.3 The College may change or amend the academic regulations, course structure or syllabi at any time, and the changes or amendments made shall be applicable to all students with effect from the dates notified by the college.

16.4 Where the words "he", "him", "his", occur in the regulations, they include "she", "her", "hers".

17.0

MALPRACTICES RULES

DISCIPLINARY ACTION FOR IMPROPER CONDUCT IN EXAMINATIONS

	Nature of Malpractices/Improper conduct	Punishment
	If the student:	
1. (a)	Possesses or keeps accessible in the examination hall, any paper, notebook, programmable calculator, cell phone, pager, palm computer, or any other form of the material concerned with or related to the subject of the examination (theory or practical) in which the student is appearing but has not made use of (material shall include any marks on the body of the student which can be used as an aid in the subject of the examination)	Expulsion from the examination hall and cancellation of the performance in that subject only.
(b)	Gives assistance or guidance or receives it from any other student orally or by any other body language methods or communicates through cell phones with any student or persons in or outside the exam hall in respect of any matter.	Expulsion from the examination hall and cancellation of the performance in that subject only of all the students involved. In the case of an outsider, he will be handed over to the police and a case is registered against him.
2	Has copied in the examination hall from any paper, book, programmable calculators, palm computers or any other form of material relevant to the subject of the examination (theory or practical) in which the student is appearing.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the student has already appeared including practical examinations and project work and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The hall ticket of the student is to be canceled.
3	Impersonates any other student in connection with the examination.	The student who has been impersonated shall be expelled from the examination hall. The student is also debarred and forfeits the seat. The performance of the original student who has been impersonated shall be canceled in all the subjects of the examination (including practicals and project work) that already appeared and shall not be allowed to appear for examinations of the remaining subjects of that semester/year. The student is also debarred for two consecutive semesters from class work and all examinations. The continuation of the course by the student is subject to academic regulations in connection with the forfeiture of the seat. If the imposter is an outsider, he will be handed over to the police and a case is registered against him.

4	Smuggles in the answer book or additional sheet or takes out or arranges to send out the question paper during the examination or answer book or additional sheet, during or after the examination.	Expulsion from the examination hall and cancellation of the performance in that subject and all the other subjects the student has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The student is also debarred for two consecutive semesters from class work and all University examinations. The continuation of the course by the student is subject to academic regulations in connection with the forfeiture of the seat.
5	Uses objectionable, abusive, or offensive language in the answer paper or in letters to the examiners or writes to the examiner requesting him to award pass marks.	Cancellation of the performance in that subject.
6	Refuses to obey the orders of the chief superintendent/assistant–superintendent/any officer on duty or misbehaves or creates disturbance of any kind in and around the examination hall or organizes a walkout or instigates others to walk out, or threatens the officer-in-charge or any person on duty in or outside the examination hall of any injury to his person or to any of his relations whether by words, either spoken or written or by signs or by visible representation, assaults the officer-in-charge, or any person on duty in or outside the examination hall or any of his relations, or indulges in any other act of misconduct or mischief which result in damage to or destruction of property in the examination hall or any part of the college campus or engages in any other act which in the opinion of the officer on duty amounts to use of unfair means or misconduct or has the tendency to disrupt the orderly conduct of the examination.	In the case of students of the college, they shall be expelled from examination halls and cancellation of their performance in that subject and all other subjects the student(s) has (have) already appeared in and shall not be permitted to appear for the remaining examinations of the subjects of that semester/year. The students also are debarred and forfeit their seats. In the case of outsiders, they will be handed over to the police, and a police case is registered against them.
7	Leaves the exam hall taking away answer script or intentionally tears off the script or any part thereof inside or outside the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all the other subjects the student has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The student is also debarred for two consecutive semesters from class work and all Semester End Examinations. The continuation of the course by the student is subject to the academic regulations in connection with the forfeiture of seat.

8	Possesses any lethal weapon or firearm in the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the student has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The student is also debarred and forfeits the seat.
9	If a student of the college, who is not a student for the particular examination or any person not connected with the college indulges in any malpractice or improper conduct mentioned in clauses 6 to 8.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the student has already appeared including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year. The student is also debarred and forfeits the seat. Person(s) who do not belong to the college will be handed over to the police and, a police case will be registered against them.
10	Comes in a drunken condition to the examination hall.	Expulsion from the examination hall and cancellation of the performance in that subject and all other subjects the student has already appeared for including practical examinations and project work and shall not be permitted for the remaining examinations of the subjects of that semester/year.
11	Copying is detected based on internal evidence, such as, during valuation or during special scrutiny.	Cancellation of the performance in that subject and all other subjects the student has appeared for including practical examinations and project work of that semester/year examinations.
12	If any malpractice is detected which is not included in clauses 1 to 11, it shall be reported to the Dean-Academics for further action to award suitable punishment.	

*****

CVR COLLEGE OF ENGINEERING

II B.Tech. Computer Science and Engineering

I Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22HS203	Computer Oriented Statistical Methods	BS	3	1	4	40	60	100	3
2	22IT201	Digital Electronics	ES	3	0	3	40	60	100	5
3	22EE205	Basic Electrical & Electronics Engineering	ES	3	0	3	40	60	100	7
4	22CS201	Discrete Mathematics	PC	3	0	3	40	60	100	9
5	22CS202	Object Oriented Programming through Java	PC	3	0	3	40	60	100	11
Practicals										
6	22EE233	Electrical & Electronics Engineering Lab	ES	0	2	1	40	60	100	13
7	22CS231	Object Oriented Programming through Java Lab	PC	0	3	1.5	40	60	100	14
8	22DT231	Data Visualization Lab	PC	0	3	1.5	40	60	100	16
			Total	**15**	**9**	**20**	**320**	**480**	**800**	
			Total Hours	**24**						
9	22HS231/281	Gender Sensitization Lab	MC	0	2	0	100	0	100	17

Service Courses of II B.Tech. I Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS201	Discrete Mathematics (CSE- AI&ML, CSE-CS, CSE-DS and IT)	PC	3	0	3	40	60	100	9
2	22CS202	Object Oriented Programming through Java (CSE-AI&ML, CSE- CS, CSE-DS and IT)	PC	3	0	3	40	60	100	11
3	22CS203/253	Database Management Systems (CSE-AI&ML, CSE-CS and CSE- DS)	PC	3	0	3	40	60	100	23
Practicals										
1	22CS231	Object Oriented Programming through Java Lab (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	0	3	1.5	40	60	100	14
2	22CS232/282	Database Management Systems Lab (CSE-AI&ML, CSE-CS and CSE-DS)	PC	0	2	1	40	60	100	31

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

HS: Humanities & Sciences **BS: Basic Sciences** **ES: Engineering Sciences**
PC: Professional Core **MC: Mandatory Course**

CVR COLLEGE OF ENGINEERING

II B.Tech. Computer Science and Engineering

II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS251	Computer Organization and Architecture	PC	3	0	3	40	60	100	19
2	22CS252	Advanced Data Structures through Java	PC	3	0	3	40	60	100	21
3	22CS253/203	Database Management Systems	PC	3	0	3	40	60	100	23
4	22CS254	Software Engineering	PC	3	0	3	40	60	100	25
5	22IT252	Operating Systems	PC	3	0	3	40	60	100	27
Practicals										
6	22CS281	Advanced Data Structures through Java Lab	PC	0	2	1	40	60	100	29
7	22CS282/232	Database Management Systems Lab	PC	0	2	1	40	60	100	31
8	22CS283	Operating System and Assembly Language Programming Lab	PC	0	2	1	40	60	100	34
9	22CS284	Real-Time/Field-Based Research Project	PC	0	4	2	50	0	50	
			Total	**15**	**10**	**20**	**370**	**480**	**850**	
			Total Hours	**25**						
10	22HS251/201	Constitution of India	MC	3	0	0	100	0	100	36

Service Courses of II B.Tech. II Semester Course Structure

II Semester Course Structure

Regulations: R22-CBCS **With effect from the Academic Year 2023-24 Onwards**

S No.	Course Code	Name of the Course	Category	Periods per Week		Credits	Scheme of Examination Maximum Marks			Page No.
				L	T/P/D		Internal	External	Total	
1	22CS252	Advanced Data Structures through Java (CSE-AI&ML, CSE- CS, CSE-DS and IT)	PC	3	0	3	40	60	100	21
Practicals										
1	22CS281	Advanced Data Structures through Java Lab (CSE-AI&ML, CSE-CS, CSE-DS and IT)	PC	0	2	1	40	60	100	29

Note: Lecture Hours (L), Tutorials (T), Practicals (P), Drawing (D) & Credits (C)

HS: Humanities & Sciences **BS: Basic Sciences** **ES: Engineering Sciences**
PC: Professional Core **MC: Mandatory Course**

Course Code:22HS203

COMPUTER ORIENTED STATISTICAL METHODS

(Common to CSE & IT)

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: By studying this course students will be able

1. To introduce the concepts of Probability and statistics.
2. To learn how to apply Probability and Statistics to solve engineering problems.
3. To keep a balance between theory and methodology.
4. To show the applications of Probability and Statistics in engineering with examples.
5. To learn the concepts of stochastic processes and Markov chains.

Unit I – Probability

Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence and the Product Rule, Bayes' Rule.

Random Variables and Probability Distributions: Concept of a Random Variable, Discrete Probability Distributions, Continuous Probability Distributions.

Unit II – Expectation and Discrete Distributions

Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and variances of Linear combinations of Random Variables, Chebyshev's Theorem, the definition of bivariate random variables, Concepts and applications of correlation and regression.

Discrete Probability Distributions: Binomial Distribution, Poisson distribution.

Unit III – Continuous Probability Distributions and Sampling Distributions

Continuous Probability Distributions: Uniform Distribution, Normal Distribution, Areas under the Normal curve, Applications of the Normal Distribution, Normal approximation to Binomial Distribution.

Fundamental Sampling Distributions: Random Sampling, Some important statistics, Sampling Distribution, Sampling Distribution of Mean, variance (Chi-square, t, F distribution) (without proof), Central Limit Theorem.

Unit IV – Sample Estimation & Tests of Hypotheses

Sample Estimation: Introduction, Statistical Inference, Classical Methods of Estimation, Prediction Intervals.

Statistical Hypotheses: General Concepts, Null Hypothesis, Alternate Hypothesis, Type-I and Type-II errors, critical region, level of Significance, Power of the Test, One-tailed and Two-tailed tests, calculation of p-value.

Single sample: Large sample tests concerning single mean and single proportion. Two samples: Large sample tests concerning two means, two proportions and large sample tests concerning variances.

Unit V – Stochastic Processes and Markov Chains

Introduction to Stochastic Processes- Markov process, Transition Probability, Transition Probability Matrix, First order and Higher order Markov processes, n-step transition probabilities, Markov chain, Steady state condition, Markov analysis.

Course Outcomes: At the end of the course, the student acquires the ability to

CO 1 : Compute Probabilities using theorems in probability and probability distributions.

CO 2 : Formulate and solve problems involving random variables and apply statistical methods for analyzing experimental data.

CO 3 : Apply the tools in Probability and Statistics in Engineering.

CO 4 : Apply the concept of estimation of parameters and testing of hypothesis about the parameters to case studies.

CO 5 : Correlate the concepts of one unit to the concepts of other units.

Textbooks:

1. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, 9th Edition Pearson Publishers, 2011.
2. Fundamentals of Mathematical Statistics, S C Gupta and V K Kapoor, Khanna Publications, 2002.
3. Operations Research, S.D.Sharma, Kedarnath and Ramnath Publishers, Meerut, Delhi, 2002.

References:

1. Fundamentals of Probability and Statistics for Engineers, James T.T. Soong, John Wiley & Sons, Ltd, 2004.
2. Probability and Statistics for Engineers and Scientists, Sheldon M Ross, 5th edition, Academic Press, 2014.
3. Probability and Statistics for Engineers, Miller and Freund’s, 8th Edition, Pearson Educations, 2015.

Course Code: 22IT201

DIGITAL ELECTRONICS

(Common to CSE and IT)

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 impart basic properties of Boolean algebra and to simplify Boolean functions.
2. To design fundamental components of a computer such as multiplexers and registers using combinatorial and sequential circuits.
3. To impart skill set to translate Boolean functions into modular programmable components.

UNIT - I: Boolean Algebra and Logic Gates

Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic**.**

Basic Definitions, Axiomatic definition of Boolean Algebra, Basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms, other logic operations, Digital logic gates, Integrated circuits.

UNIT - II: Gate – Level Minimization

The map method, Four-variable map, Five-Variable map, product of sums simplification, Don’t-care conditions, NAND and NOR implementation using Two-level implementations, Exclusive – OR function, Quine-McClusky Method – Row and Colum Dominance.

UNIT - III: Combinational Logic

Combinational Circuits, Analysis procedure, Design procedure, Binary Adder-Subtractor, Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers.

UNIT - IV: Sequential Logic

Sequential circuits, latches, Flip-Flops, Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure**.** Registers, shift Registers, Ripple counters, synchronous counters, other counters.

UNIT – V: Memories and Asynchronous Sequential Logic

Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction, Read-only memory, Programmable logic Array, programmable Array logic, Sequential Programmable Devices.

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

CO 1	:	Understand and master different number systems and realize the binary operations of Boolean algebra using logic gates.
CO 2	:	Solve gate-level minimization problems using K-map and Quine-Mc Cluskey methods.
CO 3	:	Analyze a given combinational circuit and design a new optimized circuit for a given specification.
CO 4	:	Analyze a given sequential circuit and design an optimal circuit to implement a memory element or a counter.
CO 5	:	Realize Programmable logic elements used in the design of processors and embedded systems.

Textbooks:

1. Digital Design – M. Morris Mano, 5th edition, Pearson Education/PHI, 2013.
2. Digital Principles and Applications Albert Paul Malvino Donald P. Leach, 8th edition, TATA McGraw-Hill, 2014.

References:

1. Fundamentals of Logic Design, Roth, 5th edition, Thomson.
2. Switching and Logic Design, C.V.S. Rao, Pearson Education
3. Digital Principles and Design – Donald D.Givone, Tata McGraw-Hill, Edition.
4. Fundamentals of Digital Logic and Microcomputer Design, 5th edition, M. Rafiquzzaman JohnWiley.

Course Code: 22EE205

BASIC ELECTRICAL AND ELECTRONICS ENGINEERING

(Common to CSE & IT)

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 : Learn the basics of various fundamental laws, analysis of electrical circuits, and study the nature of ac quantities.

2 : To study the construction, functioning of different types of electrical machines and their performance.

3 : To study the working and applications of various electronic devices.

Unit I - Network analysis

Ohm’s law, basic circuit components, power and energy calculations, types of elements, Kirchhoff’s laws. Resistive, inductive andcapacitive networks, series and parallel circuits, star delta and delta star transformation. Mesh and Nodal Analysis. Principle of superposition, Simple problems.

Alternating Quantities

Basic definitions: frequency, average values and RMS values of alternating currents and voltage, form factor and peak factor.

Unit II - DC Machines

Construction of dc machines, DC Generator- working and principle of operation, EMF equation, types of DC Generators. DC Motor - working and principle of operation, types of dc motor, torque equation, losses and efficiency calculations. Simple problems.

Unit III - Transformers

Principles of operation, Constructional Details, Ideal Transformer and Practical Transformer, Losses, Transformer Test, Efficiency and Regulation Calculations

3-Phase Induction Machines: Principle of operation of induction motor, Torque, Slip, Slip – torque characteristics – applications.

Unit IV - Diode Applications

Rectifiers – Half wave, Full wave and Bridge rectifiers, introduction to filters, Capacitor filter, Zener Diode characteristics Voltage regulationusing Zener Diode, Varactor Diode.

Unit V - Transistor Biasing and Amplifiers

Need for Biasing, operating point, Bias stability, DC load line, Fixed Bias, Voltage divider Bias, Principal of operation of CE Amplifier.

Components of LT Switchgear:

Switch Fuse Unit (SFU), MCB, Earthing, Types of Batteries.

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

CO 1 : Identify basic circuit components and solve basic electrical and electronic problems using different principles.
CO 2 : Understand the construction and working of different types of DC machines and calculate the losses and efficiency.
CO 3 : Understand the Construction and working principle of AC machines and their applications in real time.
CO 4 : Analyze and design different types of diodes and rectifiers.
CO 5 : Study different protection devices and basics transistor circuits.

Textbooks:

1. Principles of Electrical Engineering and Electronics, V.K.Mehta, 3rd edition, S.Chand & Co, 2014.
2. Electronics Devices and Circuits, S Salivahanan & N SureshKumar, 4th edition, McGraw-Hill, 2017.

References:

1. Basic Electrical and Electronics Engineering-S. K.Bhattacharya, 2nd edition, Pearson Education India, 2017.
2. Basic Electrical Engineering, D Kothari, I Nagrath, 3rd edition, McGraw-Hill Education, 2009.
3. Basic Electrical & Electronics Engineering, J. B. Gupta, S.K.Kataria & Sons, 2013.
4. Electronics Devices and Circuits, R.L. Boylestand and Louis Nashelsky, 9th edition, Pearson/Prentice Hall, 2006.

Course Code: 22CS201

DISCRETE MATHEMATICS

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

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 inculcate mathematical thinking and problem-solving skills in Logic, Relations, and Inferences.
2. To expose students to a wide variety of mathematical concepts that are used in Computer Science based on Number Theory and Combinatorics.
3. To represent real-world problems over Graphs and solve similarity and traversal related problems.

Unit I - Mathematical Logic

Statements and notations, connectives, Well Formed Formulas, Truth tables, tautology, equivalence implication, Normal forms, Predicative logic, Quantifiers, universal quantifiers, Free & Bound variables.

Unit II – Inference and Relations

Rules of inference, Consistency, Proof by contradiction, Automatic Theorem proving, and Applications.

Properties of binary Relations, Equivalence, Transitive closure, Compatibility & Partial ordering Relations, Lattice and its properties, Hasse Diagram. Recursive functions, and Applications.

Unit III - Algebraic structures

Algebraic systems Examples and general properties, semi-groups and Monoids, Groups, subgroups, Homomorphism & Isomorphism, and Applications

Unit IV - Elementary Combinatorics and Recurrence Relations

The principle of inclusion and exclusion, Binomial Coefficients, Binomial & Multinomial theorems, Pigeon hole principle, and its applications.

Generating Functions-Generating Functions of sequences, calculating the coefficient of generating function and applications. Recurrence Relations- Homogenous and non-homogeneous, and their solutions.

Unit V - Graph Theory

Basic Concepts, Isomorphism and Subgraphs, Planar Graphs, Euler’s Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

Course Outcomes:

CO 1 **:** Apply formal logic proofs and/or informal, but rigorous, logical reasoning to evolve theoretical proofs to real problems, such as predicting the behavior of software or solving problems such as puzzles.

CO 2 **:** Apply the logical notations to define and reason about fundamental mathematical concepts such as sets, and relations and exercise the guidelines for constructing valid arguments. A representation of a partially ordered set such as a lattice as a directed graph.

CO 3 **:** Define Group properties and construct simple functions that preserve the algebraic structures over groups.

CO 4 **:** Solve counting problems efficiently by applying the principle of inclusion and exclusion and solve recurrence relations.

CO 5 **:** Characterize edge preserving similarity between two graphs and verify the Eulerian property of graphs.

Textbooks:

1. Discrete Mathematical Structures with Applications to Computer Science, J.P. Tremblay, and R. Manohar, Tata McGraw-Hill Publishing Company, 2008.
2. Discrete Mathematics for Computer Scientists & Mathematicians, J.L. Mott., A. Kandel and T.P. Baker, 2nd edition, Prentice Hall, 2009.

References:

1. Discrete Mathematics and its Applications, Kenneth H.Rosen, 7th edition, TMH, 2015.
2. Discrete and Combinatorial Mathematics- An Applied Introduction, Ralph P. Grimaldi, 5th edition, Pearson Education, 2008.
3. Elements of Discrete Mathematics – A computer Oriented Approach, C L Liu, and D P Mohapatra, 3rd edition, Tata McGraw-Hill, 2008.

Course Code: 22CS202

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

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

Instruction	:	3Periods / 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 and apply various object-oriented programming features like abstraction, encapsulation, inheritance and polymorphism to solve various computing problems using Java language.
2. To identify, define and implement exception handling and multi-threading mechanisms in application domains.
3. To design and develop GUI applications using AWT & Swings and Understanding of the new features.

Unit I - Introduction to Java and Building Blocks of Java

Basics of Java- History/Background of Java, Java Buzzwords, Java Virtual Machine and Byte code, Java Environment setup, Java Program structure, Data Types, Variables- Scope and Life Time, Operators, Expressions, Type Conversions and Type casting, Conditional statements and Control statements, Simple Java Programs, javac and java command flags.

OOP Concepts –I: Encapsulation- Classes and Objects, Classes: Class structure, class components, Objects: Object declaration, Reference variables, Constructors - default Constructor, Parameterized Constructors, Constructor overloading, this keyword and its uses, arrays concept, static modifier, access modifiers, Wrapper classes.

Methods -Passing parameters to methods – Passing primitive types and Passing Objects, getters and setters, Method Overloading, Command line arguments, garbage collection-java.lang.System.gc(), finalize(). **String Handling -** String class, String APIs, String Buffer and String Builder classes.

Unit II - OOP Concepts –II

Inheritance- Inheritance concept, super class and subclass relationship, Object class, principle of substitution, effect of access modifiers on inheritance. Usage of super (field, method, constructor) and final(field, class, method) keywords.

Polymorphism- method overriding, Dynamic method dispatch, Abstract classes and Interfaces - Abstract classes - concept, usage, Interfaces – declaration, implementation, components of an interface, extending interfaces.

Packages – package access, CLASSPATH, package access rules, sealed classes, hidden classes, Introduction to Java standard library and Java documentation.

Unit III - Dealing exceptions and I/O

Exception Handling: Fundamentals of exception handling, benefits of exception handling, Exception types, Termination or presumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, exception hierarchy, throw, throws and finally, built in Exceptions, Custom exceptions, Throwable Class.

Java I/O–Byte streams, character streams, Scanner class, Console class, Serialization and Serializable interface, File class.

Unit IV – Multithreading and Modules

Multithreading-Fundamentals, Thread Life Cycle, Ways of creating threads - Thread class and Runnable interface, Thread priorities, creating multiple threads, core methods of Thread class, Thread Synchronization, inter thread communication.

Annotations- Annotation Basics, specifying a Retention Policy, the Annotated Element Interface, Using Default Values, Marker Annotations, Single – Member Annotations.

Modules: Module Basics-module, exports, require, transitive, java.base and the Platform Modules, Unnamed Module, Specific Module.

Unit V - GUI Development

AWT - Basics of GUI Programming, Event handling – Delegation event model, event sources, event listeners, event classes, adapter classes: nested classes and interfaces, anonymous inner classes handling keyboard and mouse events.

Swing- MVC Architecture, Containers, components, layout managers, frames and windows, panels, buttons, checkboxes, radio buttons, combo boxes, lists, labels, color choosers, file choosers, text fields, text areas, tool tips.

Course Outcomes:

At the end of the course, student should be able to

CO 1	**:**	Design and implement object-oriented concepts like encapsulation, abstraction and data hiding using programming constructs offered by java language.
CO 2	**:**	Realize the power of inheritance, interfaces, and packages.
CO 3	**:**	Understand and demonstrate the concepts of exception handling and java io streams.
CO 4	**:**	Demonstrate knowledge and understanding of multi-threading, annotations, and modules in Java.
CO 5	**:**	Design and develop java applications using AWT & Swings and make use of the advanced features for providing solutions to real world problems.

Textbooks:

1. Java: The Complete Reference, Herbert Schildt, 11th edition, McGraw-Hill Education, Oracle Press, 2019.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, O'Reilly Media, 2005.

References:

1. Core Java Volume I- Fundamentals, Cay S. Horstmann and Gary Cornell, 9th edition, Prentice Hall, 2012.
2. Core Java Volume II- Advanced Features, Cay S. Horstmann and Gary Cornell, 9th edition, Prentice Hall, 2013.

Course Code: 22EE233

ELECTRICAL AND ELECTRONICS ENGINEERING LAB

(Common to CSE & IT)

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Course Objectives:

1 : Students analyze various electrical parameters and different theorems.
2 : Students can understand working and operation of different electrical machines.
3 : Students learn working and applications of various semiconductor devices.

List of the Experiments:

Part-A

(Any 5 experiments from below given list)

1. Verification of Superposition principle with Resistive load
2. Verification of voltage division and current division in series & parallel circuits
3. Magnetization characteristics of D. C Shunt generator.
4. Swinburne's Test on DC shunt machine (Predetermination of efficiency of a given DC shunt machine working as motor and generator).
5. O. C & S. C tests on Single-phase transformer. (Predetermination of efficiency and regulation at given power factors and determination of Equivalent circuit)
6. Measurement of RMS and average values of AC quantities.

Part-B

(Any 5 experiments from below given list)

1. Zener diode characteristics
2. Rectifier without filters (HWR and FWR)
3. Rectifier with filters (HWR and FWR)
4. Bridge rectifier
5. Design of Voltage divider bias Circuit
6. Frequency response of Common Emitter Amplifier

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

CO 1 : Understand and verify superposition principle.
CO 2 : Determine currents and voltages in parallel and series circuits.
CO 3 : Analyze the performance characteristics of various electrical machines.
CO 4 : Design Rectifier circuits.
CO 5 : Understand the basic transistor biasing techniques.

Course Code: 22CS231

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

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

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 set up the necessary environment for running java applications.
2. To implement the basic concepts of object-oriented programming.
3. To implement the practical aspects of exception handling, multithreading mechanisms and Java I/O.
4. To be able to design and implement applications using GUI components.

Lab Problems:

1. a. Write a program to implement the different types of operators, to perform the following tasks: comparison of values, simple arithmetic, bit–wise operations.
   b. Write a program to check and print the grade of a student when the score is given as an integer. Use a switch statement. Rewrite the program to use a sequence of if-else statements.
   c. Write a program to demonstrate the command-line arguments.
2. a. Write a program to demonstrate the task of overloading of constructors and methods.
   b. Write a program to understand the concept of type casting.
3. a. Use an array of integers and find the sum and average of the elements of that array.
   b. Practice further programs on the usage of arrays.
4. a. Write a program to utilize both standard and custom packages. The program should reflect the usage of packages in a correct manner, along with the purpose of access modifiers.
   b. Write a program to use gc() method of both System and Runtime classes. Experiment with other methods of those classes.
5. a. write a program using the hierarchy of employees in a university.
   b. Write a program to understand polymorphic invocation of methods, while overriding the methods. Use an employee base class and manager sub class; override the computeSalary() method to illustrate the concept.
   c. Develop an application that uses inheritance. Use the class Account and then subclass it into different account types. Then making use of Customer and Employee classes to develop the application to reflect the nature of banking operations. Use minimum operational sequence.
6. a. Demonstrate the use of abstract classes. Write a Person abstract class and then subclass that into Student and Faculty classes. Use appropriate fields and methods.
   b. Write a program to demonstrate the usage of interfaces.
7. a. Write a program to understand the full capability of String class. Implement as many methods as required. Consult API documentation to read through the methods.
   b. Write programs using StringBuffer and StringBuilder library classes.
8. a. Write a program to demonstrate the usage of try and associated keywords. Introduce bugs into the program to raise exceptions and then catch and process them.
   b. Learn how to create and use custom exceptions.
   c. Experiment on using various methods of Throwable, Exception classes and Practice on chaining the exceptions.
9. a. Using byte streams, write a program to both read from and write to files.
   b. Using FileReader and FileWriter, write a program to perform file copying and any other suitable operations.
   c. Write a Java Program that displays the number of characters, lines and words in a text file.
10. a. Use the classes StringTokenizer, StringReader and StringWriterto write a program to find the capabilities of these classes.
    b. Write a program to demonstrate enumerations and usage of Assertions.
    c. Demonstrate assertions through simple programs.
11. a. Write programs to illustrate the use of Thread class and Runnable interface.

b. Write a program to show the assignment of thread priorities.
c. Write a program to synchronize threads. Use Producer and Consumer problem to illustrate the concept.

12. a. Create simple advanced calculator, which checks whether a number is prime, calculates the sum of 'N' prime numbers, checks whether a number is even, and calculates the sum of 'N' even and odd numbers using modules.
    b. Write a java program to perform the operations: sort and search on an array of integers and define the following: i. Simple Junit testcases ii. Multiple testcases iii. Suite test
13. a. Write a program to design a frame and control its various display properties.
    b. Write a program to understand the Keyboard and Mouse Events using adapter classes.
14. a. Write a program to demonstrate any layout manager. Use a suitable application.
    b. Write a GUI based application to demonstrate the usage of various javax.swing components and the corresponding event handling techniques.

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

CO 1	**:**	Implement object-oriented concepts like encapsulation, data hiding, and abstraction using programming constructs offered by java language.
CO 2	**:**	Develop java programs to realize the power of inheritance, interfaces, and packages.
CO 3	**:**	Develop java programs to demonstrate the concepts of exception handling and I/O streams.
CO 4	**:**	Implement java applications using a multithreading mechanism and understand the power of modules.
CO 5	**:**	Use graphical user interfaces to create Frames for providing solutions to real-world problems.

References:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, O'Reilly Media, 2005.
3. https://junit.org/junit5/docs/current/user-guide/#running-tests-junit-platform-runner.

Course Code: 22DT231

DATA VISUALIZATION LAB

(Common to CSE, CSE-CS, CSE-DS & IT)

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. Understand the various types of data, apply and evaluate the principles of data visualization.
2. Acquire skills to apply visualization techniques to a problem and its associated dataset.

List of Experiments:

1. Understanding Data types & creating respective charts at Univariate, Bivariate and Multivariate.
2. Creating dashboards for effective data visualization.
3. Acquiring and plotting data.
4. Statistical Analysis – such as Multivariate Analysis, PCA, LDA, Correlation regression and analysis of variance.
5. Financial analysis using Clustering, Histogram and HeatMap.
6. Time-series analysis – stock market.
7. Visualization of various massive dataset - Finance - Healthcare - Census – Geospatial.
8. Visualization on streaming dataset (Stock market dataset, weather forecasting).
9. Market-Basket Data analysis-visualization.
10. Text visualization using web analytics

Course Outcomes:

CO 1: Identify the different data types, visualization types to bring out the insight.

CO 2: Relate the visualization towards the problem based on the dataset to analyze and bring outvaluable insight on a large dataset.

CO 3: Demonstrate the analysis of a large dataset using various visualization techniques and tools.

CO 4: Identify the different attributes and showcasing them in plots. Identify and create various visualizations for geospatial and table data.

CO 5: Ability to create and interpret plots using R/Python.

References:

1. Robert Spence "Information visualization – Design for interaction", Pearson Education, 2nd edition, 2007.
2. Alexandru C. Telea, "Data Visualization: Principles and Practice," A. K. Peters Ltd, 2008.

Course Code: 22HS231/281

GENDER SENSITIZATION LAB

(Mandatory Course) (Common to all Branches)

Instruction	:	2 Periods/week	Sessional Marks	:	100
Credits	:	0			

Course Description:

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology,philosophy, political science, anthropology and media studies – to examine cultural assumptions about sex, gender, and sexuality.

This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex and gender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

Course Objectives:

1. : To develop students' sensibility with regard to issues of gender in contemporary India.
2. : To provide a critical perspective on the socialization of men and women
3. : To introduce students to information about some key biological aspects of genders
4. : To expose the students to debates on the politics and economics of work
5. : To help students reflect critically on gender violence and to support a sustainable gender-equal society.

Unit I - Understanding Gender

Introduction: Definition of Gender-Basic Gender Concepts and Terminology-Exploring Attitudes towards Gender-Construction of Gender-Socialization: Making Women, Making Men -Preparing for Womanhood. Growing up Male. First lessons in Caste.

Unit II Gender Roles and Relations

Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles-Gender Roles and Relationships Matrix-Missing Women-Sex Selection and Its Consequences-Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary

Unit III - Gender and Labour

Division and Valuation of Labour-Housework: The Invisible Labor- "My Mother doesn't Work." "Share the Load."-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and Unaccounted work.

-Gender Development Issues-Gender, Governance, and Sustainable Development- Gender and Human Rights-Gender and Mainstreaming

Unit IV - Gender-Based Violence

The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No!-Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: *"Chupulu".*

Domestic Violence: Speaking Out: Is Home a Safe Place? -When Women Unite [Film]. Rebuilding Lives.Thinking about Sexual Violence Blaming the Victim-"I Fought for my Life…."

Unit V Gender and Culture

Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature- Gender Development Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals

Mary Kom and Onler. Love and Acid just do not Mix. Love Letters. Mothers and Fathers. Rosa Parks-The Brave Heart.

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

CO 1 : Students will have developed a better understanding of important issues related to gender in contemporary India.

CO 2 : Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender.

CO 3 : Students will attain a finer grasp of the biological spheres of gender in our society and how to counter it.

CO 4 : Students will acquire insight into the gendered division of labor and its relation to politics and Economics.

CO 5 : Students will develop a sense of appreciation for women in all walks of life and contribute to establish an egalitarian society.

Textbook:

1. *Towards a World of Equals: A Bilingual Textbook on Gender,* A. Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu, Telugu Academy, Telangana Government, 2015.

Course Code: 22CS251

COMPUTER ORGANIZATION AND ARCHITECTURE

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

Pre-requisite: A Course on **"Digital Electronics**".

Course Objectives:

1. The purpose of the course is to introduce principles of computer organization and basic architectural concepts.
2. It begins with the basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3. Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

Unit I

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design, and Computer Architecture.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro-operations, shift micro-operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input–Output and Interrupt.

Unit II

Microprogrammed Control: Control memory, Address sequencing, microprogram example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, and Program Control.

Unit III

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.
Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating–point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

Unit IV

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

Unit V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor arbitration, Inter-processor communication and synchronization, Cache Coherence.

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

CO1	:	Understand the basics of instruction sets and their impact on processor design.
CO2	:	Demonstrate an understanding of the design of the functional units of a digital computer system.
CO3	:	Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory
CO4	:	Design a pipeline for consistent execution of instructions with minimum hazards.
CO5	:	Recognize and manipulate representations of numbers stored in digital computers.

Textbooks:

1. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI, 2014.

References:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th edition, Mc Graw Hill, 2011.
2. Computer Organization and Architecture – William Stallings 6th edition, Pearson/PHI, 2002.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4th edition, PHI/Pearson, 2013.

Course Code: 22CS252

ADVANCED DATA STRUCTURES THROUGH JAVA

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

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 importance of generic programming, Java’s collection framework and functional programming.
2. To implement various basic data structures like stacks, queues, linked lists etc. using user defined generic classes and java’s collection classes.
3. To learn various data structures for implementing dictionaries.

Unit I - Generics and Functional Programming

Generics: Introduction to Generics, simple Generics examples, Generic Types, Generic methods, Bounded Type Parameters and Wild cards, Inheritance & Sub Types, Generic super class and sub class, Type Inference, Restrictions on Generics.

Functional Programming: Functional Interfaces – Function, BiFunction, Predicate, and Supplier, Lambda Expression Fundamentals, Block Lambda Expressions, Passing Lambda Expressions as Arguments, Lambda Expressions and Exceptions, Variable Capture, Method References.

Unit II - 1D and 2D Collections & Stream API

1D Collection: 1D Collection Interfaces: Collection, Set, List, NavigableSet, SortedSet, Queue, Deque. 1D Collection Classes-Hash Set, Linked HashSet, TreeSet, ArrayList, LinkedList.

2D Collection:2D Collection Interfaces-Map, NavigableMap, SortedMap, 2D Collection Classes-HashMap, LinkedHashMap, TreeMap.

Stream API: Stream basics, Stream Interface, Intermediate operations – map(), filter(), distinct(), sorted(), limit(), skip(), Terminal operations – forEach(), reduce(), collect(), min(), max(), count().

Unit III - Dictionaries

Introduction: Dictionary definition, Dictionary ADT.

Dictionaries Implementation-I:

Linear List Representation: Basics of linear list, implementation of sorted list using user defined generic classes and, LinkedList Collections class.

Hashing: basics, closed hashing – linear probing, quadratic probing, double hashing, rehashing, extendible hashing and their implementation, open hashing-separate chaining and its implementation using user defined generic classes.

Binary Search Trees: definition and basics, implementation of operations-searching, non-recursive traversals, insertion and deletion using user defined generic classes.

Unit IV - Dictionaries Implementation-II

AVL Tree: definition, the height of an AVL tree, representation, operations-rotations, insertion, searching, deletion and, their implementation using Java’s Collection framework.

Red-Black Binary search trees: definition, insertion, deletion, and search operations.

Unit V - B-Trees and Priority Queues

B-Tree: B-Tree of order m, the height of a B-Tree, searching, insertion, and deletion operations.

Priority Queue: definition, max and min heaps, realizing priority queues using heaps, operations-insertion, deletion, and their implementation using user-defined generic classes, heap sort and its implementation using user-defined generic classes.

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

CO 1	**:**	Realize the power of generics and functional programming in java.
CO 2	**:**	Understand Java's Collection class hierarchy and also know the power of data processing using streams.
CO 3	**:**	Implement dictionaries using linear lists, hashing & binary search tree and compare their performances.
CO 4	**:**	Implement dictionaries using an AVL tree and red, black tree.
CO 5	**:**	Understand the advantages of B-trees and Priority Queues.

Textbooks:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Data Structures and Problem-Solving Using Java, Mark A. Weiss, 4th edition, Pearson Education, 2009.

References:

1. Data Structures and Algorithms in Java, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th edition, Wiley Publications, 2014.
2. Data Structures, Algorithms, And Applications in Java, Sartaj Sahni, 2nd edition, Universities Press, 2005.
3. Data Structures: Abstraction and Design Using Java, Elliot B. Koffman, Paul A. T.Wolfgang. 2nd second Edition, Wiley publications, January 2010.
4. Head First Java, Kathy Sierra and Bert Bates, 2nd edition, OREILLY publications, 2005.

Course Code: 22CS253/203

DATABASE MANAGEMENT SYSTEMS

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

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 the role of database management systems in an organization.
2 : To represent real-world scenarios using E-R diagrams.
3 : To model the database using relations avoiding redundancies.
4 : To learn transaction management and concurrency protocols to ensure data consistency.
5 : To understand the database file organization system and database recovery techniques.

Unit I - Introduction to DBMS

History of DBMS, Concepts, and overview of DBMS, Data models - ERmodel, Relational model, Levels of Abstraction in DBMS, Database Languages, Architecture of DBMS, Data Base Users and Administrators.

ER-Model

Database design and ER model, ER modeling Constructs, Additional features of ER Model, Class Hierarchies, Aggregation, Conceptual Design with ER model, Case study: ER design for Large Enterprises.

Unit II - Relational Algebra and Calculus

Introduction to the relational model, Logical Database Design- ER to Relational, Relational Algebra - Selection and Projection, Set operations, Renaming, joins, Examples of Relational Algebra Relational Calculus- Tuple relational Calculus, Domain relational calculus.

Introduction to Structured Query Language

Form of Basic SQL Query, Examples of Basic SQL Queries, Introduction to Nested Queries, Correlated Nested Queries, Set Comparison Operator-Aggregate Operators, NULL values and Comparison using Null values, Logical connectivity’s – AND, OR and NOT, OUTER Joins, Disallowing NULL Values.

Unit III - PL/SQL

Data Types, Declaration of Variables, Strings, Control Conditional Statements, Functions, Procedures, Cursors, and Triggers.

Schema Refinement

Introduction to schema refinement, Problems caused by decomposition, Functional dependencies (FDs) and reasoning about FDs, Normal Forms (NF) – 1NF, 2NF, 3NF and BCNF, Properties of Decomposition, Schema Refinement in Data Base Design, Case studies using Normal Forms

Unit IV - Transaction Management

Transaction concept & state, Implementation of atomicity and durability, Concurrent executions of a transaction, Serializability and Recoverability, Implementation of Isolation, Testing for serializability, Lock-Based Protocols, Graph-Based Protocol, Timestamp-Based Protocols, Validation-Based, Protocols, Multiple Granularity.

Unit V – Database File Organization and Recovery

Data Base File Organization

Data on External storage, File Organization and Indexing, Cluster Indexes, Primary, and secondary indexes, Index data structures, Hash-based indexing - Static hashing and Extensible Hashing, Tree based indexing - Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index structure.

Database Recovery
Recovery and Atomicity, Log-based Recovery, and Recovery with the concurrent transaction.

Course Outcomes:

At the end of the course, the students should be able to

CO 1 : Demonstrate an understanding of database management system components and features. Design E-R Model torepresent real-world database application scenarios.

CO 2 : Demonstrate a mathematical approach towards querying a database using relational algebra and relational calculus and implement using SQL.

CO 3 : Convert E-R Model to a relational Model and design a proper relational database while eliminating anomalies.

CO 4 : Demonstrate the role of transaction management and concurrency control protocols.

CO 5 : Demonstrate an understanding of database file organization and recovery of the database in case of crashes.

Textbooks:

1. Database System Concepts, A.Silberschatz, H.F. Korth,S.Sudarshan, 6th edition, McGraw-Hill, 2006.
2. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.

References:

1. Fundamentals of Database Systems, Ramez Elmasri, Shamkant B.Navathe, 7th edition, Pearson Education, 2008.
2. Database Systems: The Complete Book by Hector Garcia- Molina, Jeffery D.Ullman, Jennifer Widom, 2nd Edition, Pearson Education, 2008.
3. Database Management System Oracle SQL and PL/SQL, P.K.Das Gupta, 2nd edition, PHI, 2013.

Course Code: 22CS254

SOFTWARE ENGINEERING

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. The aim of the course is to provide an understanding of the working knowledge of the techniques for understanding the requirements, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, UML diagrams, software testing, software process/product metrics, risk management, and quality management.

Unit - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. **A Generic view of process**: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). **Process models**: The waterfall model, Spiral model and Agile methodology.

Unit - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

Unit - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

Unit - IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Metrics for Process and Products: Software measurement, metrics for software quality.

Unit - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. **Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

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

CO 1 : Understand the need for software engineering and use of different software process models for different types of projects.

CO 2 : Translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).

CO 3 : Identify and apply appropriate software architectures to carry out high level design of a system and be able to carry out detailed design using different UML diagrams.

CO4 : Develop a strategic approach to testing and debugging. Will have experience and/or awareness of testing and debugging problems. Be able to apply metrics to assess software quality.

CO5 : Identify software risks and apply RMMM. Be able to conduct formal technical reviews on artifacts during different stages of software development to improve quality of the artifacts.

Textbooks:

1. Software Engineering, A practitioner’s Approach- Roger S. Pressman, 8th edition, McGraw-Hill International Edition, 2014.
2. Software Engineering- Sommerville, 10th edition, Pearson Education, 2017.

References:

1. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson,Pearson Education, 2005.
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley, 2019.
3. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies, 2004.
4. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education, 1999.

Course Code: 22IT252

OPERATING SYSTEMS

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

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

Prerequisites:

1. A course on "Computer Programming and Data Structures".

Course Objectives:

1. Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection).
2. Introduce the issues to be considered in the design and development of operating system.
3. Introduce basic Unix commands, system call interface for process management, inter-process communication and I/O in Unix.

Unit I – Operating System Introduction

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls.

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads. Process related system calls – fork, exit, wait and exec.

Unit II - CPU Scheduling, Process Management and Synchronization

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware and Software, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

Unit III - Interprocess Communication Mechanisms and Deadlocks

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

Unit IV - Memory Management and Virtual Memory

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

Unit V - File System Interface and Operations

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Disk scheduling algorithms, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

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

CO1 : Understand the role of Operating System with its function and services.
CO2 : Compare various algorithms used for CPU scheduling and apply various concepts related to concurrency and synchronization to solve problems.
CO3 : Understand the inter process communication mechanism and resolve deadlock in a multi-programmed environment.
CO4 : Understand the concepts of virtual memory and how it is realized in systems
CO5 : Differentiate and Demonstrate file systems, directory structures and their implementation issues.

Textbooks:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th edition, John Wiley, 2017.
2. Advanced programming in the UNIX environment, W.R. Stevens, 3rd edition, Pearson education, 2013.

References:

1. Operating Systems- Internals and Design Principles, William Stallings, 5th edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Croley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education

Course Code: 22CS281

ADVANCED DATA STRUCTURES THROUGH JAVA LAB

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

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To implement generic programming and Java's collection framework.
2. To apply Java's collection framework for implementing basic data structures like stacks, queues, linked lists, etc.
3. To understand the concepts of functional programming, lambda expressions and streams.
4. To implement dictionaries using advanced data structures like Binary search trees, and AVL trees.

Lab problems:

1. Write a java program to demonstrate the use of bounded type parameters and wild card arguments.
2. Write a java program that returns the value of pi using the lambda expression.
3. Write a java program that takes a string as parameter and calculates the reverse of the string using lambda expression.
4. Write a java program to implement iterators on Array List and LinkedList.
5. a) Implement a Generic stack to deal with Integer, Double and String data using user-defined arrays and linked lists.
   b) Implement a Generic queue to deal with Integer, Double and String data user-defined arrays and linked lists.
6. a) Write a Java program to implement Generic stack using Array List Collection class.
   b) Write a Java program to implement Generic stack using LinkedList Collection class.
7. a) Write a Java program to implement Generic queue using ArrayList Collection class.
   b) Write a Java program to implement Generic queue using LinkedList Collection class.
8. Write a Java program to demonstrate the use of the following Collection classes.
   a. HashSet b. LinkedHashSet c. TreeSet
9. Write a java program to create a class called Person with income, age, and name as its members. Read set A of persons from a user and compute the following sets:
   i) Set B of persons whose age > 60
   ii) Set C of persons whose income < 10000 and iii) B ∩ C
10. Write a Java program to demonstrate the use of the following Collection classes.
    a. HashMap b. LinkedHashMap c. TreeMap
11. Create a class Product(id, name, price, type, rating) and perform the following operations using stream:
    i) Find all the products having rating between 4 and 5.
    ii) Find first n products having price > 10000.
    iii) Find the number of products under each type(map containing type and count).
    iv) Find average rating of products with type = "Electronics".
12. Write a Java program to implement Sorted Chain.
13. Write a Java program to implement Separate Chaining
14. Write a Java program to implement Linear Probing.
15. Implement BST using Collection API, and use recursive procedures to implement inOrder, preOrder and postOrder traversals.
16. Implement AVL tree using Collection API.
17. Implement priority queues with max Heap tree using Collection API.
18. Implement heap sort with max Heap tree using Collection API.

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

CO 1 **:** Understand the power of generics and functional programming.

CO 2 **:** Implement hashing, sets, stacks and queues using collection classes in java.util package and process the data using streams.

CO 3 **:** Implement dictionaries using various data structures like sorted list, and hashing.

CO 4 **:** Implement dictionaries using various height-balanced trees and also analyze the advantages and disadvantages of height-balanced trees.

CO 5 **:** Understand the importance of Priority Queues and their applications.

References:

1. Java: The Complete Reference, Herbert Schildt, 10th edition, McGraw-Hill Education, Oracle Press, 2017.
2. Data Structures and Problem Solving Using Java, Mark A. Weiss, 4th edition, Pearson Education, 2009.
3. Data Structures and Algorithms in Java, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, 6th edition, Wiley Publications, 2014.

Course Code: 22CS232/282

DATABASE MANAGEMENT SYSTEMS LAB

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

Instruction	:	2 Periods / week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	--	Semester End Examination	:	60 Marks
Credits	:	1.0	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To understand the relational model.
2. Analyze database requirements and determine the entities involved in the system and their relationship to each other.
3. Understand logical design of the database modeling concepts such as E-R diagrams.
4. Demonstrat SQL DML/DDL commands to insert and manipulate the database.
5. Understand procedures, functions and triggers in PL/SQL.

Database Description: This lab enables the students to practice the concepts learnt in the subject DBMS by developing a database for an example - **Boat reservation by the sailor** and -**employee data maintenance in an organization** whose description is as given below. The student is expected to practice the designing, developing and querying a database in the context of reserving a boat and employee data maintenance. Students are expected to use - MySq database.

"Boat reservation by the sailor" is a schema with several boats which could be reserved depending on color and availability on a particular day. The sailor reserves the boat on a particular day y registering himself with a rating. The sailor is identified by sailor id, boats are identified by boat id and reservation is uniquely identified by sailor id, boat id and day.

"Employee data maintenance in an organization": In any organization, we need to maintain the data of employees categorized into department as per the salary. The scheme contains employee, department and sal grade tables which are identified by employee id, department id and range of salary respectively.

1. E-R Model

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.

Boat reservation by the sailor:

Entities:

1. SAILORS
2. BOATS
3. RESERVES

PRIMARY KEY ATTRIBUTES:

1. SID (SAILOR ENTITY)
2. BID (BOATS Entity)
3. SID,BID,DAY (RESERVES ENTITY)

Employee data maintenance in an organization Entities:

1. EMPLOYEE
2. DEPT
3. SALGRADE

PRIMARY KEY ATTRIBUTES:

1. EID (EMPLOYEE ENTITY)
2. DID (DEPT Entity)
3. LOWSAL AND HIGHSAL (SALGRADE ENTITY)

2. Concept design with E-R Model

Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc. wherever required for

1) Boat reservation by the sailor

2) Employee data maintenance in an organization

3. Relational Model

Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality.Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi valued, and Derived) have different way of representation.

SAILORS

SID	SNAME	RATING	AGE

EMPLOYEE

EID	ENAME	DID	SAL	DESIGNATION	MGRNUM	DOJ	AGE

4. Normalization

Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical orstructural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

Perform do the second and third normal forms for sailors and Employee databases if required.

5. Installation of Mysql and practicing DDL commands

Installation of MySql. In this week student will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. Students will also try truncate, rename commands etc.

6. Practicing DML commands

DML commands are used to for managing data within schema objects. Some examples:

1) SELECT - retrieve data from the a database

2) INSERT - insert data into table

3) UPDATE - updates existing data within a table

4) DELETE - deletes all records from a table, the space for the records remain

7. Querying - I

In this week students are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

8. Querying - II

Students are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

9. Triggers

In this week students are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

10. Procedures

In this session students will learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

11. Cursors

In this week students will learn to declare a cursor that defines a result set. Open the cursor to establish the result set. Fetch the data into local variables as needed from the cursor, one row at a time. Close the cursor when done.

Course Outcomes:

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

CO1 : Analyze database requirements and determine the entities involved in the system and their relationship to each other.
CO2 : Design E-R Model to represent database application scenarios.
CO3 : Convert/transform the E-R Model to relational tables, populate relational database and formulate SQL queries on data.
CO4 : Improve the database design by normalization.
CO5 : Implement PL/SQL procedures, function, triggers and cursors.

References:

1. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, 3rd edition, TMH, 2003.
2. Introduction to SQL, Rick F.VanderLans, 4th edition, Pearson education, 2007.
3. Oracle PL/SQL, B.Rosenzweig and E.Silvestrova, 2nd edition, Pearson education, 2002.

Course Code: 22CS283

OPERATING SYSTEM AND ASSEMBLY LANGUAGE PROGRAMMING LAB

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

Course Objectives:

1. To provide an understanding of the design aspects of operating system concepts through simulation
2. To introduce system call interface for process management, inter-process communication and I/O in Unix
3. To enable students gain hands on experience on Assembly Language Programming on 8086

List of Experiments:

1. Implement shell commands such as cp, ls, chmod, ls -ls using the I/O system calls of UNIX/LINUX operating system.
   (open, read, write, close, fcntl, seek, stat, opendir, readdir)
2. Write C programs to simulate the following CPU Scheduling algorithms:
   a) FCFS b) SJF.
3. Write C programs to simulate the following CPU Scheduling algorithms:
   a) Round Robin b) priority.
4. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
5. Write C programs to illustrate the following IPC mechanisms.
   a) pipes b) FIFOs
6. Write C program to illustrate the Message Queues IPC mechanism.
7. Write C program to illustrate the Shared Memory IPC mechanism.
8. Write a C program to implement the Producer–Consumer problem using semaphores using UNIX/LINUX system calls (pthread library API is optional).
9. Write an ALP in 8086 add, subtract and multiply two 16-bit unsigned numbers.
10. Write an ALP in 8086 to implement ASCII Adjust and decimal adjust instructions.
11. Write an ALP to pack two digits into a Byte.
12. Write an ALP to Count number of 1’s and number of 0’s present in the binary representation of a given number.
13. Implement the following string manipulation functions using appropriate registers.
    a) Copy a string b) Lower to upper case c) Reverse a string d) Palindrome.
14. Write an ALP to Count no of even and odd numbers from the given array of numbers.
15. Write a program to check whether a given number is Positive or Negative number.
16. Write an ALP to sort the given array of numbers.
17. Write C programs to simulate Paging memory management techniques.
18. Write C programs to simulate Segmentation memory management techniques.

Note: Programs 1 to 16 are mandatory and 17, and 18 are optional.

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

CO1 : Write programs using I/O System calls for implementing file operations

CO2 : Simulate and implement operating system concepts such as scheduling, deadlock management, and memory management.

CO3 : Implement and realize the semantics of synchronous and asynchronous Inter - Process communication models.

CO4 : Demonstrate the memory segmentation and implement the programming model of Intel 8086 processor

CO5 : Realise the data and string manipulation instructions

References:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th edition, John Wiley, 2006.
2. Advanced Programming in the Unix Environment, W.R.Stevens, 2nd edition, *Pearson* education, 2015.
3. Operating Systems – Internals and Design Principles, William Stallings, 5th edition, Pearson Education/PHI, 2005.
4. Advance Microprocessors and Peripherals – A.K.Ray and K.M.Bhurchandani, TMH, 3rd edition, 2013.
5. Microprocessors and Interfacing – D.V.Hall, TMGH, 2nd edition, 2006.

Course Code: 22HS251/201

CONSTITUTION OF INDIA

(Mandatory Course) (Common to all Branches)

Instruction	:	3 Periods/week	Sessional Marks	:	100
Credits	:	0			

Course Objectives: Students will be able to

1 : Understand the history and making of the Indian Constitution.
2 : Recognize the Philosophy of the Indian Constitution and Preamble
3 : Identify the importance of fundamental rights as well as fundamental duties.
4 : Understand the functioning of organs of governance and local administration
5 : Learn composition and activities of Election Commission and institutional bodies.

Unit I

History of Making of the Indian Constitution: The meaning of constitutional Government, the roots of the constituent Assembly of India, Composition of the proposed constituent Assembly. History of Drafting Committee

Unit II

Philosophy of the Indian Constitution: Salient features of Indian Constitution, Preamble of the Constitution. Contours of Constitutional Rights & Duties - Fundamental Rights-Right to Equality, Right to Freedom, Right against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Directive Principles of State Policy and Fundamental Duties.

Unit III

Organs of Governance: Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions.

Unit IV

Local Administration: District’s Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Panchayat raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO Zila Panchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

Unit V

Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State Election Commission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.

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

CO 1 : Understand and explain the significance of Indian Constitution as the fundamental law of the land.
CO 2 : Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India. Exercise his

fundamental rights in proper sense at the same time identifies his responsibilities in national building.

CO 3 : Analyze the organs of governance and District's Administration head

CO 4 : Analyse the Local Administration: District's and Village Administration

CO 5 : Understand Election Commission Process and Institutional Bodies for the welfare of SC/ST/OBC and women.

Textbooks:

1. The Constitution of India, 1950 (Bare Act), Government Publication, 2015.
2. Dr. S. N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.
3. M. P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

Course code: 22CS301

COMPUTER NETWORKING

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 communication reference models and to understand the characteristics of the transmission media.
2 : To solve the optimal route establishment problems for data delivery using relevant metrics.
3 : To serve data at the end point level and to ensure reliable data delivery mechanisms.
4 : To model secured exchange of high-level data between two applications.

Unit I – Introduction

Uses of Computer Networks, Network Hardware, Network Software, Reference Models, Example Networks. **Physical Layer** - Guided Transmission Media, Wireless Transmission.

Unit II - Data Link Layer

Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, example of data link protocols.
Medium Access Control Sub-Layer -The Channel Allocation Problem, Multiple Access Protocols, Ethernet, Wireless LANs.

Unit III - Network Layer

Network Layer Design Issues, Routing Algorithms, Congestion Control Algorithms, Quality of Service. Internetworking, Network Layer in the Internet.

Unit IV - Transport Layer and Application Layer

Transport Service, Elements of Transport Protocols, Internet Transport Protocols- User Datagram Protocol, Transmission Control Protocol.
Application Layer -Domain Name System, Electronic Mail, World Wide Web. Simple Network Management Protocol.

Unit V – Information Security

Cryptography, security services, message confidentiality, message integrity, message authentication, digital signature.

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

CO 1 : Appreciate communication reference models along with PDUs and design a physical network based on the topological specifications using various communication media.
CO 2 : Design solutions for logical link layer transmission and control errors, control the medium access patterns through an established methodology.
CO 3 : Solve the optimal routing problems for static and dynamic networks and realizing various QoS parameters.
CO 4 : Demonstrate the significance of various applications using TCP and UDP protocol.
CO 5 : Implement CIA security mechanisms, and Network Security protocols.

Text Books:

1. Computer Networks, Andrew S Tanenbaum,4th Edition, Pearson Education, 2011.
2. Data Communications and Networking, Behrouz A. Forouzan, 5th Edition, TMH, 2009.

References:

1.Computer Networking: A Top-Down Approach, James F. Kurose, and K.W. Ross7th Edition, Pearson Education, 2017.
2. Introduction to Data Communications and Networking, W. Tomasi, Pearson Education, 2009.
3. Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education, 2008.

Course Code: 22CS302

ALGORITHMS DESIGN AND ANALYSIS

(Common to CSE, IT)

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 emphasize upon the demands of real-world problems in engineering solutions
2 : To make students conversant with the various paradigms of algorithms
3 : To handcraft the performance analysis of designed solutions
4 : To take students through various optimization principles of ill-posed problems

Unit I – Fundamentals of algorithm analysis

Introduction- Definition of algorithm, algorithmic problem solving, pseudo code for expressing algorithms. Asymptotic notations- O, Ω, and θ notations. Performance analysis: Time and space complexity: count, tabular methods, examples on non-recursive, recursive algorithms. Recursive algorithms and recurrence relations - ToH problem, Amortized analysis.

Unit II - Divide and Conquer

Control abstraction, binary search algorithm and its complexity, Merge sort, its complexity, quick sort, its complexity. Graph traversals: Depth first search (dfs), breadth first search (bfs), articulation points, bi-connected components. Disjoint Sets: Union and Find.

Unit III - Greedy paradigm

Control abstraction, fractional knapsack problem, job sequencing problem, minimum cost spanning tree: Prim’s algorithm, Kruskal’s algorithm, Single source shortest path algorithm, Huffman coding.
Pattern matching algorithms: Knuth-Morris Pratt algorithm and brute force algorithm.

Unit IV - Dynamic programming and Backtracking

Dynamic programming **-** Control abstraction, Multistage Graphs, OBST, Travelling salesperson problem, reliability design, 0/1 knapsack problem.
Backtracking - n-queens problem, Graph coloring, Sum of subsets problem.

Unit V - Branch and Bound, Complexity Theory

Branch and Bound: General method, Applications: Travelling sales person problem, 0/1 knapsack problem, LC branch and bound solution, FIFO branch and bound solution, Game trees, Heuristics for search space reduction: alpha-beta pruning.
Np-hard and NP-complete problems: basic concepts, non-deterministic algorithms, NP-hard and NP-complete classes, Cook’s theorem.

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

CO 1 : Analyze worst-case running times using asymptotic analysis of algorithms.
CO 2 : Describe the divide-and-conquer paradigm and Synthesize divide-and-conquer algorithms.
CO 3 : Define optimization problems and solve them through various greedy policies
CO 4 : Describe the dynamic-programming paradigm and synthesize dynamic-programming algorithms and analyze them.
CO 5 : Reduce the size of search space of the optimization problems by applying backtracking and branch and bound tools. Appreciate the Non-Deterministic modeling of algorithms.

Textbooks:

1. Fundamentals of algorithms, E. Horowitz and S.Sahni, 2nd edition, Galgotia Publications, 2010
2. Introduction to algorithms, T.H.Cormen, C.E.Leiserson, R.L.Rivest, and C.Stein, 2nd edition, PHI/Pearson Education, 2001.

References:

1. Introduction to Design and Analysis of Algorithms, A strategic approach, R C T Lee, Hang and TT Sai, TMH, 2012
2. Data structures and Algorithm Analysis in C++, Allen Weiss, 2nd edition, Pearson Education, 2002.
3. Design and Analysis of Algorithms, Aho, Ullman and Hopcroft, Pearson Education, 1974.
4. Algorithms, Richard Johnson Baugh, and Marcus Schaefer, Pearson Education, 2004.

Course Code: 22IT301

WEB TECHNOLOGIES

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

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

Prerequisites:

1. Must have knowledge in HTML 5 and CSS 3.
2. Must be knowledgeable on Java Technology.
3. Must be knowledgeable on any RDBMS.

Course Objectives:

1 : To learn a framework to create responsive web designing.
2 : To learn the client-side script and validations along with asynchronous programming.
3 : To introduce XML and work with data storage and interactivity using Java.
4 : To introduce Server-side programming with Java Servlets.
5 : To learn sending Dynamic Response from server using JSP.

Unit I – Working with CSS and its Framework

Introduction to CSS: Syntax structure, using style sheets, Box model. **CSS3:** Grid, Flexbox. Responsive Web Design using Media Queries, use of viewport, Transition, Animation.

CSS Framework: Bootstrap.

CSS Framework: Bootstrap (local and CDN usage, containers, 12 – column grid system, commonly used controls – Typography, Nav, Navbar, Carousel, Button, Card, Modal dialog, Table, forms, Breadcrumbs).

Unit II – Client- Side Scripting using JavaScript

JavaScript: Introduction to JavaScript, Data types, var, let, const., Control statements, Operators, Functions, fat arrows, Arrays, Objects, Destructuring, Strings, Date Objects, Events, DOM Manipulations, Regular Expressions.

Introduction to jQuery: Syntax, Selectors, Events, Effects.

Unit III – Data storage and manipulation

XML: Syntax, namespaces, DTD, Schema, XML Document Parsing.

JDBC: Design of JDBC, JDBC Configuration, Working with JDBC Statements, Scrollable and UpdatableResultSets, Rowset, MetaData, Transactions.

Unit IV – Server-side Scripting using Servlets

Web servers: An introduction to Web Servers, Web application structure and deployment in Tomcat. MVC Architecture, Servlet Technology: Servlets, Servlet lifecycle, The Servlet API packages and class and interface hierarchy, basic servlet program template, Handling requests and responses using form parameters, using ServletContext and ServletConfig objects, using initialization parameters (both context and config level), Session management (Cookies, Http Session, URL Rewriting, HiddenForm fields).

Unit V – Dynamic Response using JSP

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

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

CO 1	:	Build a custom website with HTML, CSS, and Bootstrap
CO 2	:	Demonstrate JavaScript and its asynchronous nature of execution
CO 3	:	Implement the Database Connectivity and Component Technologies like Beans
CO 4	:	Develop and deploy Servlet based web applications
CO 5	:	Develop Server-side programming using JSP

Textbooks:

1. Teach Yourself HTML, CSS, and JavaScript All in One, Julie C. Meloni, Jennifer Kyrnin, Sams 3rd Edition, Pearson Publication, 2019.
2. Head First Servlets and JSP, Bryan Basham, Kathy Sierra and Bert Bates, O'Reilly Media, 2ndEdition, 2008.
3. Core Java Volume II—Advanced Features, Cay S. Horstmann, 10th Edition, Prentice HallPublications, 2017

References:

1. Responsive Web Design with HTML5 and CSS3, Ben Frain, Second Edition, Packt Publishing,2015.
2. Beginning HTML, XHTML, CSS, and JavaScript, Jon Duckett, Wiley Publishing, Inc., 2010.
3. Core Servlets and JSPs, Martin Hall and Larry Brown, Volume I and II, Pearson.
4. E – Resource: https://www.w3schools.com/html/
5. E – Resource: https://developer.mozilla.org/en-US/docs/Learn/JavaScript
6. E – Resource: https://getbootstrap.com/

Course code: 22CS303

DATA SCIENCE

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 data science and descriptive analysis of data.
2	:	To develop models to perform association, regression, and classification on data.
3	:	To develop a comprehensive understanding of various Data analysis methods.

Unit I – Introduction to Data Science

Data science, Characteristics of Bigdata, Different steps in Data science process, Types of Data analytics.
Descriptive Analysis: Data Types and Scales, Types of Data Measurement Scales, Measures of Central Tendency, Measures of Variation, Similarity, and dissimilarity measures,
Data preprocessing: Data Cleaning, Data Integration, Data Transformation.

Unit II – Exploratory Data Analysis

Hypothesis testing: t-Test, z-Test, Chi-Square-Test.
Analysis of Variance (ANOVA): One-way, Two-way.
Multivariate Analysis: Mean Vector, Covariance, Correlation and Precision Matrices, Multivariate Data, Parameter Estimation, Estimation of Missing Values, Multivariate Normal Distribution.
Dimensionality Reduction: Principal Component Analysis and Multi-Dimensional Scaling.

Unit III – Predictive Analysis-I

Simple linear Regression: Model Building, Estimation of Parameters, Interpret coefficients, Validation of model, Outlier analysis. Bias, variance, and trade-off, Gradient descent, over and under fitting models.

Multiple linear regression: Ordinary Least Squares Estimation, Model building, Validation of model, coefficients of Multiple determination R^2 and adjusted R^2, Multicollinearity and Variance Inflation factor, Ridge, Lasso regression.

Unit IV- Predictive Analysis-II

Classification: Basic concepts, K – Nearest Neighbor classifier, Support Vector Machines, Nonlinear boundaries, Kernel functions, Bayes classifier/Graph analytics, Model Evaluation and Selection, Techniques to Improve Classification Accuracy.

Logistic Regression: Binary Logistic Regression, Estimation of Parameters, Interpretation of Parameters, Model Diagnostics, Classification Table, Sensitivity, and Specificity.

Unit V – Prescriptive Analysis

Forecasting Techniques: Time series data, Techniques, and accuracy, Moving average method, Single, double, triple exponential smoothing, Regression model for forecasting, Auto-Regression models, ARIMA Process.
Graph Analytics: Path analysis, Connectivity analysis, Community analysis, Centrality analysis, Social-Network Graphs, Communities and Clusters, Betweenness, The Girvan-Newman Algorithm.

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

CO 1 : Preprocess the data and can perform descriptive analysis
CO 2 : Implement multivariate data analysis and Hypothesis testing
CO 3 : Able to apply Regression models to solve real world problems.
CO 4 : Develop classification methods to solve problems
CO 5 : Understand prescriptive analysis using forecasting techniques and Graph analytics

Textbooks:

1. Business Analytics, U. Dinesh Kumar, Wiley publications, 2017.
2. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeff Ullman, Cambridge University Press, 2016.

References:

1. Data Mining: Concepts and Techniques, Jiawei Han and Micheline Kamber, 4rd edition, Morgan Kaufmann Publishers, 2023.
2. Introduction to Machine Learning, Ethem Alpaydin, Francis Bach , 3rd edition Adaptive Computation and Machine Learning series 2014.
3. Principles of Data Science, Sinan Ozdemir, Packt publications, 2016.
4. Data Science in Theory and Practice, Maria Cristina Mariani, Osei Kofi Tweneboah and Maria Pia Beccar-Varela, John Wiley and Sons publishers, 2022.
5. Introduction to Data Mining, Pang-Ning Tan, Michael Stein batch, Vipin Kumar, Pearson publications, 2016.

Course Code: 22CS304

COMPUTER GRAPHICS

(Professional Elective - I)
(Common to CSE & IT)

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 : The aim of this course is to provide an introduction of fundamental concepts and theory of computer graphics.

2 : It presents the important drawing algorithm, polygon fitting, clipping and 2D transformation curves and an introduction to 3D transformation.

3 : To expose the students to the current and emerging technologies such as OpenGL and visualize 2D and 3D objects.

4 : It provides the basics of Animation techniques interface which allows students to develop programming skills in CG.

Unit I – Introduction

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations, Frame buffers, and input devices.
Output primitives: Points and lines, line drawing algorithms (Bresenham’s and DDA Algorithm), mid- point circle and ellipse algorithms
Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

Unit II - 2-D Geometrical transforms

2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems
2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland algorithms, Sutherland – Hodgeman, polygon clipping algorithm.

Unit III - 3-D Object representation

Polygon surfaces, quadric surfaces, spline representation, Hermite curve,Bezier curve and B-Spline curves. Basic illumination models, surface rendering methods and polygon rendering methods.

Unit IV - 3-D Geometric transformations

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations,composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Unit V – Computer Animation

Computer animation: Design of animation sequence, general computer animation functions, raster animation, key frame systems, motion specifications.

Visible surface detection methods: Classification, back-face detection, depth-buffer, BSP-tree methods and area sub-division method K-d Tree, OCTree and Ray-casting method.

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

CO1 : Know the application areas of computer graphics, overview of graphics systems and output primitives.

CO2 : Apply 2D geometric transforms, 2D viewing using transformation matrices.

CO3 : Acquire familiarity with the relevant mathematics of computer graphics.

CO4 : Apply 3D Geometric representations and transformations.

CO5 : Be able to design basic graphics application programs, including animation.

Textbooks:

1. Computer Graphics C version, Donald Hearn and M. Pauline Baker, Pearson Education, 2002.
2. A Programming approach: Computer Graphics, Steven Harrington, Second Edition, Tata Mc Graw Hill, 1987.

References:

1. "Computer Graphics Principles & practice", Foley, Van Dam, Feiner and Hughes, second edition in C, Pearson Education, 2003.
2. "Computer Graphics", Zhigang Xiang, Roy Plastaock, second edition, Schaum's Outlines, 2000.
3. "Procedural elements for Computer Graphics", David F Rogers, 2nd edition, Tata Mc Graw hill, 2001.

Course Code: 22CS305

DISTRIBUTED DATABASES

(Professional Elective - I)
(Common to CSE, IT)

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 enrich the previous knowledge of database systems and expose the need for distributed database technology to confront the deficiencies of centralized database systems.
2 : To introduce basic principles and implementation techniques of distributed database systems.
3 : To make the students understand the concurrency and serializability in transaction management system.
4 : To acquire the knowledge on distributed database reliability and parallel database design.
5 : To equip students with principles and knowledge of parallel and object-oriented databases.

Unit **I -**

Introduction: Distributed Data Processing, Distributed Database System, Promises of DDBSs,Problem areas.
Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDBMS Architecture.
Distributed Database Design: Alternative Design Strategies, Distribution Design Issues, Fragmentation, Allocation.

Unit II

Query Processing and Decomposition: Query processing objectives, characterization of queryprocessors, layers of query processing, query decomposition, localization of distributed data.
Distributed query Optimization: Query optimization, centralized query optimization, distributed queryoptimization algorithms.

Unit III

Transaction Management: Definition, properties of a transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms and algorithms, time-stamped and optimistic concurrency control algorithms, deadlock management.

Unit IV

Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local and distributed reliability protocols, site failures, and network partitioning.
Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

Unit V

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query processing.
Object Oriented Data Model: Inheritance, object identity, persistent programming languages,persistence of objects, comparison OODBMS and ORDBMS

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

CO1	:	Understand theoretical and practical aspects of distributed database systems.
CO2	:	Identify various issues related to the development of distributed database systems.
CO3	:	Understand serializability and concurrency control in distributed databases.
CO4	:	Design a fault tolerant distributed system and able to run parallel query processing.
CO5	:	Understand the design aspects of object-oriented database systems and related development.

Textbooks:

1. Principles of Distributed Database Systems, M. Tamer OZSU and Patuck Valduriez: Pearson Edn. Asia, 2001.
2. Distributed Databases: Principles and Systems, Stefano Ceri and Giuseppe Pelagatti: McGraw Hill 2017.

References:

1. "Database Systems: The Complete Book", Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom Second Edition, Pearson International Edition, 2008

Course Code: 22CS306

MICROPROCESSORS AND INTERFACING

(Professional Elective - I)

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 appreciate pipeline implementation Architecture and Programming of 8086.
2 : To interface real-world peripherals with the processor and controller.
3 : To design synchronous and asynchronous communication models.
4 : To design Microcontroller architecture, Memory organization, Instruction set with simple programs.
5 : To Learn and Understand the Real time control using programming serial communication and programming timers and counters.

Unit I - 8086 Architecture

8086 Architecture–Functional Diagram, Register Organization, Memory Segmentation, Signal Descriptions of 8086, Minimum and Maximum Modes, Physical Memory Organization, Timing Diagrams, Addressing Modes 0f 8086, Instruction set of 8086, Assembler Directives.

Unit II - Memory and I/O Interfacing

SRAM Interfacing and DRAM Interfacing, 8255 PPI Architecture, Various Modes of Operation of 8255 and Interfacing with 8086, Displays, Keyboard Interfacing, Interfacing Analog to Digital converter: ADC 0808/0809, Interfacing Digital to Analog converter: DAC 0800.

Unit III - Interrupts and Serial Communication Interface

Interrupt Structure of 8086, Vector Interrupt Table, Interrupt Service Routine, Interrupt Controller 8259 Architecture and interfacing with 8086. Introduction to DOS and BIOS Interrupts.
Serial Communication Standards, Serial Data Transfer Schemes, 8251 USART Architecture and Interfacing, RS-232.

Unit IV - 8051 Microcontroller

Overview of 8051 Microcontroller, Architecture, I/O ports, Memory Organization, Addressing Modesand Instruction Set of 8051, Simple Programs.

Unit V - 8051 Real Time Control

Interrupts, timer/counter and serial communication, Programming Timer Interrupts, Programming external hardware interrupts, programming the serial communication interrupts, Programming 8051 timers and counters.

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

CO 1 : Gain knowledge about pipelined processor 8086 and understand timing diagrams.
CO 2 : Interface the processor with peripheral devices.
CO 3 : Gain knowledge about interrupt structure and serial communication of 8086 microprocessor.
CO 4 : Master the 8051 architecture and programming
CO 5 : Implementing various real time controls like timers, interrupts, serial communications in 8051 micro-controller.

Textbooks:

1. Advanced Microprocessors and Peripherals, A. K. Ray and K. M. Bhurchandani, 2nd Edition, Tata McGraw-Hill, 2006.
2. The 8051 Microcontroller, Kenneth J. Ayala, 3rd Edition, Cengage Learning, 2010.

References:

1. Microprocessor and Interfacing, D.V. Hall, 2nd Edition, Tata McGraw-Hill, 2006.
2. Micro Computer system: 8086/8088 Family Architecture, Programming and Design, Liu and G. A. Gibson, 2nd Edition, Prentice Hall, 1986.
3. The 8051 Microcontroller and Embedded Systems, Muhammad Ali Mazidi and Janice Gillispie Mazidi, 2nd Edition, Pearson, 2008.

Course Code: 22CS307

PRINCIPLES OF PROGRAMMING LANGUAGES

(Professional Elective - I)

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 Learn and understand Programming Language concepts, Environments, and syntax and semantics.
2 : To Understand conceptual design of high-level language implementation using data types, expressions, and statements.
3 : To implement the subprograms, co-routines using functions, design issues, abstract datatypes
4 : To understand and implement programs with concurrency, exception handling in various programing languages C++, Java, and C#.
5 : To learn and understand the Functional and imperative languages using LISP, scripting languages, key concepts, and Case study PHP programs.

Unit I – Preliminary Concepts

Preliminary Concepts: Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments
Syntax and Semantics: General Problem of Describing Syntax and Semantics, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs.

Unit II – Data Types, Expressions and Statements

Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments, Named Constants
Data Types: Introduction, Primitive Data Types, Character String Types, User defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence
Expressions and Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

Unit III - Subprograms

Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User defined Overloaded Operators, Closures, Coroutines Implementation.
Subprograms: General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping
Abstract Data Types: The Concept of Abstraction, Introduction to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations.

Unit IV – Concurrency, Exception Handling

Concurrency: Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency.
Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

Unit V – Functional and Imperative Languages

Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages.
Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.
Scripting Language: Pragmatics, Key Concepts.
Case Study: PHP – Values and Types, Variables, Storage and Control, Bindings and Scope, Forms creation, Inheritance in PHP.

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

CO 1 : Understand the role of theoretical formalisms, such as BNF for syntax and operational and denotational semantics for semantics
CO 2 : Understand the salient features in the landscape of programming languages like Data Types, Bindings, scope, expressions etc.
CO 3 : Analyze the different ways of sub program execution and ADT implementation across different programming languages.
CO 4 : Comprehend Concurrency and Exception Handling concepts and their implementation in various programming languages.
CO 5 : Understand the basics of functional programming languages logic programming languages and scripting languages.

Textbooks:

1. Concepts of Programming Languages 10/E, Robert. W. Sebesta, Pearson Education, 2016.
2. Programming Language Design Concepts, D. A. Watt, Wiley Dreamtech, 2007.

References:

1. Programming Languages, A.B. Tucker, R. E. Noonan.2nd Edition, TMH,2007.
2. Programming Languages, K. C. Louden, 2nd Edition, Thomson, 2003
3. Introduction to Data Communications and Networking, W.Tomasi, Pearson Education, 2009.
4. Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education, 2008.

Course code: 22CS308

DIGITAL IMAGE PROCESSING AND COMPUTER VISION

(Professional Elective-I)
(Common to CSE, IT)

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 offer background knowledge about Image representations and elementary transformations.

2 : To make the students understand the essential functionality of Image Processing.

3 : To introduce various transformations and filters for image processing enhancement, Restoration Segmentation.

4 : To develop statistical intuitions for extracting and representing the image features.

Unit I – Digital Image Fundamentals

Introduction, Origin, Steps in Digital Image Processing, Components, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels.

Unit II – Image Enhancement

Spatial Domain: Basic intensity transformation functions, Histogram processing, Fundamentals of Spatial Filtering, Smoothing and Sharpening Spatial Filtering.
Frequency Domain: Basics of filtering in frequency domain, Image Smoothing using lowpass frequency domain filters and Image Sharpening using high pass Filters.

Unit III – Image Restoration

Restoration: Noise Models, Restoration in the presence of Noise only- Spatial filtering, Periodic noise reduction, Frequency domain filtering, Inverse filtering, Wiener filtering, Reconstruction from projections.
Colour Image Processing: Colour models, Basics of full Colour image processing, Colour transformations, Colour image smoothing and Sharpening,

Unit IV – Image Compression and Segmentation

Image compression: Fundamentals, Huffman coding, Arithmetic coding, Run length coding, Predictive coding.
Morphological Processing: Erosion and Dilation.
Segmentation: Point, Line and Edge detection, Thresholding, Segmentation by Region Growing, Region splitting and merging.

Unit V – Feature extraction and Computer Vision

Boundary Preprocessing: Chain Code, Polygonal Approximation, Signature.
Boundary feature descriptors: Shape Number, Fourier Descriptor, Moments.
Regional feature Descriptors: Topological Feature, Texture, Corner Detection, Rectangle detection. Scale Space and Scale Selection; SIFT, SURF; HoG

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

CO 1 : Understand the image representation and relationship between the pixels.

CO 2 : Design and develop filters for image enhancement in the Spatial and Frequency domains to improve quality of an image.

CO 3 : Implement restoration techniques to restore degraded images and Compressions techniques to reduce the size.

CO 4 : Apply Segmentation techniques to perform image into meaningful regions.

CO 5 : Demonstrate the Feature extraction methods and solve computer vision-based problems.

Textbooks:

1. Woods, Digital Image Processing, Rafael C. Gonzalez and Richard E. Fourth edition. Pearson Education, July 2018.
2. Computer Vision: A Modern Approach, Forsyth, A., D. and Ponce, J., Pearson Education, 2nd Edition, 2012.

References:

1. Pattern classification, Richard Duda. Hart and David strok, John Wiley publishers, Second Edition 2012.
2. Image Processing. Analysis and Machine Vision, Milan Sonka, Vaciav Hlavac and Roger Boie, Second Edition, Thomson learning,2001.
3. Pattern Recognition and Image Processing, Signal Processing and Communications Series, Sing-T. Bow, Second Edition, January 2002.

Course code: 22CS331

ALGORITHMS AND DATA SCIENCE LAB

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 Marks
Tutorial	:	-	Semester End Examination	:	60 Marks
Credits	:	1	Semester End Exam Duration	:	3 Hours

List of Experiments:

Algorithms

Task 1:

a) Implement Merge sort algorithm and plot its time complexity with reference to the size of the input.

b) Implement Quick sort algorithm and plot its time complexity regarding asymptotic notations (Best, average, and worst).

Task 2:

Write a program to identify the articulation points present in a graph.

Task 3:

Implement Job Sequencing with deadlines algorithm.

Task 4:

Implement Fractional Knapsack Algorithm.

Task 5:

Implement Dijkstra’s algorithm to compute the shortest path through a graph.

Task 6:

Implement OBST using dynamic programming.

Task 7:

a) Implement sum of subset algorithm

b) Implement N-queen algorithm.

Task 8:

Implement Prim’s algorithm.

Task 9:

Implement Kruskal’s algorithm.

Task 10:

Implement graph coloring algorithm.

Data Science

Task 1:

1. Introduction to various libraries of Python for analyzing the datasets.
2. Exploring the various methods of Pandas and Numpy libraries on datasets.
3. Pre-processing and Visualization of the data.

Task 2:

1. Demonstrate the dimensionality reduction technique (PCA) on suitable dataset and compare the results before and after the reduction.
2. Demonstrate the dimensionality reduction technique (RFE) on suitable dataset and compare the results before and after the reduction.

Task 3:

1. Demonstrate Apriori Frequent Item Set Mining Algorithm on supermarket dataset to list the top 10 Association rules. Comment on the performance of the algorithm for different support and confidence threshold values.
2. Demonstrate FP-Growth Frequent Item Set Mining Algorithm on supermarket dataset to list the top 10 Association rules. Comment on the performance of the algorithm for different support and confidence threshold values.

Task 4:

1. Design and Demonstrate Regression model to predict the age of a person. Evaluate the performance of the model.

Task 5:

1. Implement the Decision Tree Classification model on Soybean dataset. Estimate the accuracy of model.
2. Implement the K-Nearest Neighbor Classification Technique on iris dataset. Estimate the accuracy of the model.

Task 6:

1. Design and implement Random Forest Classification model to predict if a loan will get approved or not for a bank customer dataset. Estimate the accuracy of the model.

Task 7:

1. Implement K-Means Clustering technique on a suitable dataset. Compute the performance measures of the clustering.
2. Implement Hierarchical Clustering technique on weather dataset. Compute the performance measures of the clustering.

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

CO 1 : Solve Sorting problems using Divide and Conquer Model.

CO 2 : Solve fractional knapsack, job sequencing with deadlines and single source shortest path problem using Greedy Method.

CO 3 : Solve OBST in dynamic programming and N-queen using backtracking

CO 4 : Demonstrate Data Science python libraries and pre-process the given data.

CO 5 : Design and Develop Prediction, Classification and Clustering models for real world applications

References:

1. Fundamentals of algorithms, E. Horowitz and S.Sahni, 2nd edition, Galgotia Publications, 2010
2. Business Analytics, U. Dinesh Kumar, Wiley publications, 2017.

Course Code: 22IT331

WEB TECHNOLOGIES LAB

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

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 learn the basics of HTML elements
2 : To learn the basics of java Console and GUI based programming
3 : To introduce XML and processing of XML Data with Java
4 : To introduce Server-side programming with Java Servlets and JSP
5 : To introduce Client-side scripting with JavaScript and AJAX.

Exercises:

1. Create a web page using the advanced features of CSS: Grid, Flexbox. And apply transition and animations on the contents of the web page
2. Make the web pages created in the above experiment as responsive web page with Bootstrap Framework.
3. Validate the registration, user login, user profile and payment pages using JavaScript. Make use of any needed JavaScript objects.
4. Build a scientific calculator.
5. JavaScript Program to demonstrate working of prototypal inheritance, closure, callbacks, promises and async / await.
6. Write an XML file which will display the Book information with the following fields: Title of the book, Author Name, ISBN number, Publisher name, Edition, Price
7. Define a Document Type Definition (DTD) and XML schema to validate the above created XML Documents
8. Write a java program to establish a connection to a database and execute simple SQL queries.
9. Write a java program to demonstrate the usage of JDBC in performing various DML statements. Use prepared statements and callable statements.
10. Write a java-based application to demonstrate the Updatable and Scrollable result sets.
11. Write a java program to access meta data of the SQL database.
12. Write a program to accept request parameters from a form and generate the response.
13. Write a program to accept Servlet Config and Servlet Context parameters.
14. Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and, pwd4 respectively. Write a servlet for doing the following functionalities
    a. Create a Cookie and add these four user ids and passwords to this Cookie.
    b. Read the user id and password entered into the Login form and authenticate with the values (user id and passwords) available in the cookies. If the person is a valid user (i.e., user-name and password match) you should welcome by name (user-name) else you should display the message “You are not an authenticated user ‟.
15. Develop a servlet to demonstrate the database access and update from a database.
16. Create a servlet to implement an authentication filter mechanism.
17. Develop a servlet to implement servlet context and session listeners.
18. Write a JSP which does the following job:
    a. Insert the details of the three users who register with the web site by using registration form.
    b. Authenticate the user when he submits the login form using the username and password from the database.
19. Write a JSP to demonstrate the usage of JSP standard actions.
20. Write a JSP to show the usage of various scripting elements.
21. Design and use a custom tag library.
22. Design a simple application using both Servlets and JSPs along with database access.

Note: Programs from 1 to 14 are mandatory and Programs from 15 to 22 are optional.

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

CO 1 : To build a custom website with HTML, CSS, and Bootstrap.
CO 2 : Demonstrate JavaScript, XML, DHTML and related Technologies.
CO 3 : Implement the Database Connectivity and Component Technologies like Beans.
CO 4 : Deploy the servlet technology & API Management.
CO 5 : Construct the fundamentals of JSP.

References:

1. Beginning HTML, XHTML, CSS, and JavaScript, Jon Duckett, Wrox Publications, 2010
2. Head First Servlets and JSP, Bryan Basham, Kathy Sierra and Bert Bates, O'Reilly Media,2nd Edition, 2008.
3. Core Java: Volume II – Advanced Features, Cay Horstmann and Gary Cornell, Prentice Hall, 9th Edition, 2013 (Only Chapter 4 for Database Programming)

Course Code: 22CS332

CASE TOOLS LAB

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 role of formal specifications in project design and be able to develop such specifications.

2 : To be able to design an interface and develop a prototype for a complex software system.

3 : To understand the role of testing in the software development cycle and be capable of developing a test plan.

4 : To be aware of and able to use Computer Aided Software Engineering (CASE) tool.

LIST OF PROGRAMS:

Task 1:

ATM Case Study and Online Railway Reservation System
Phases in software development project, overview and need. Understand problems in existing systems and perform system analysis: Requirement analysis, SRS

Task 2:

Requirement analysis, SRS of both case studies

Task 3:

To perform the function-oriented design: Data flow diagrams

Task 4:

To perform the function-oriented design: Structured chart

Task 5:

To perform the user’s view analysis: Use case diagram

Task 6:

To draw the structural view diagram: Class diagram, object diagram.

Task 7:

To draw the behavioral view diagram: Sequence diagram, Collaboration diagram

Task 8:

To draw the behavioral view diagram: State-chart diagram

Task 9:

To draw the behavioral view diagram: Activity diagram

Task 10:

To draw the implementation view diagram: Component diagram.

Task 11:

To draw the implementation view diagram: deployment diagram

Task 12:

Version Control System, GIT
1. Working Locally with GIT
2. Working Remotely with GITHUB
3. Branching and Merging
4. Resolve merge Conflict
5. GIT reset and Stash operation
6. How to setup Git on Premises Hardware

Task 13:

1. Introduction about Maven project
2. Build, test and deploy a simple application in Maven.

Task 14:

Continuous Integration using Jenkins
1. Introduction of Jenkins
2. Install and setup Jenkins
3. Continuous Build and Deployment.

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

CO 1	:	Develop requirement specifications for a software problem in hand
CO 2	:	Perform functional oriented and object-oriented design
CO 3	:	Implement the concepts of object oriented to develop a real-world application
CO 4	:	Prepare test cases to rigorously test the application for ensuring quality.
CO 5	:	Integrate developed code using Jenkins to simulate a CI/CD pipeline

References:

1. Software Engineering, Ian Sommerville , 9TH Edition, 2004.
2. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino, Prentice Hall, 2003.
3. The Unified Development Process, Ivar Jacobson, Grady Booch, James Rumbaugh , Pearson Education, 1999
4. Design Patterns: Elements of Reusable Object-Oriented Software , Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides,Pearson Education, 1994.
5. Software Metrics: A Rigorous and Practical Approach, Norman E. Fenton, Shari Lawrence Pfleeger, PWs Pub, 1997.

Course Code: 22IT332

MOBILE APPLICATION DEVELOPMENT LAB

(Common to IT, CSE)

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 understand the fundamentals of the Kotlin programming language

2 : To acquire the skills of using Android Software Development Tools and categorize the different layouts.

3 : To use common Android UI components to create a basic User Interface, handle user input and Android lifecycle events.

4 : To enhance the ability to navigate across the various screens of application.

5 : To create well-designed Android applications using Kotlin that can connect to the Internet, store data.

Exercises

Task 1 : Introduction to Kotlin

1. Write a Kotlin program that takes a nullable integer as input and print its Square if it is not null, or "Input is null" otherwise
2. Implement a number guessing game in which the user is prompted to enter a number between 1 and 100 until he or she guesses correctly. After every wrong guess, the user is told whether the guess was too high or too low.
3. Create a function in Kotlin that takes a name as input and prints the greeting message. Make the message customizable and provide a default message if no custom message is provided.

Task 2: Kotlin's Object Oriented Concepts

1. Create a Kotlin application for Rolling die using classes.
2. Create a Kotlin application to demonstrate the companion objects, getter and setter properties.

Task 3 : Creation of Android Application

1. Create "Hello World" application. That will display "Hello World" in the middle of the screen in the red color with white background.
2. Write a program to demonstrate Activity Lifecycle.

Task 4: Understanding Activity and Intent

1. To understand Activity and Intent create a sample application with login module. (Check username and password), on successful login, go to next screen and on failing login, alert the user using Toast. Also pass username to next screen.
2. Create a Dice Roller Android app that has a Button to roll a dice and update the image on the screen.

Task 5 :

1. Create a program with different types of dwellings (Shelters people live in like round hut, square cabin, round tower) that are implemented as a class hierarchy.
2. Create a tip calculator app with a working Calculate button.

Task 6 :

1. Create a polished Affirmations app that uses a Recycler View to display a list of cards. Each card contains an image and affirmation text.
2. Create a dictionary app to implement navigation between screens using intents and adding an options menu.

Task 7 :

1. Create a Words app to use a single activity and multiple fragments, and navigate between fragments with the Navigation Component.
2. Create an Unscramble game app where the user can guess the scrambled words. Use Live Data for the app's data (word, word count and the score) in the Unscramble app.

Task 8 :

1. Create a Cupcake app that displays an order flow for cupcakes, allowing the user to choose the cupcake flavor, quantity, and pickup date
2. Create a cupcake ordering app that allows the user to send the order to another app and allows for canceling an order.

Task 9 :

1. Create an Android Trivia app illustrates navigation patterns and controls. The app has several components:
2. In the title screen, shown on the left in the screenshot above, the user starts the game.
3. In the game screen with questions, shown in the middle above, the user plays the game and submits their answers.

Task 10 :

1. The navigation drawer, shown on the right above, slides out from the side of the app and contains a menu with a header. The drawer icon opens the navigation drawer. The navigation-drawer menu contains a link to the About page and a link to the rules of the game.

Task 11 :

1. Create the Guess the Word app, beginning with starter code. Guess The Word is two-player charades-style game, where the players collaborate to achieve the highest score possible.

Task 12 :

1. Build an Inventory app that saves inventory items into the SQLite database.

Task 13

1. Build an app that tracks sleep quality using database. The app uses a database to store sleep data over time. The app has two screens, represented by fragments.
2. The first screen, shown on the left, has buttons to start and stop tracking. The second screen, shown on the right, is for selecting a sleep-quality rating.

Task 14 :

1. Create an app called Mars Real Estate, which shows properties for sale on Mars. This app connects to a web service to retrieve and display the property data, including details such as the price and whether the property is available for sale or rent

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

CO 1	:	Compare and contrast the language fundamentals of Kotlin.
CO 2	:	Use Android Layouts and Connect Views with data through data binding.
CO 3	:	Illustrate the Android Lifecycle mechanism and application architecture.
CO 4	:	Develop the navigation patterns and display collections of data using recycler View.
CO 5	:	Collaborate data persistence in the application, and use APIs to connect to internet, to store and retrieve data.

Textbooks:

1. Android Programming with Kotlin for Beginners, John Horton, Packt, 2019

References:

1. Kickstart Modern Android Development with Jetpack and Kotlin, Catalin Ghita, PACKT PUBLISHING LIMITED, 2022.
2. https://developer.android.com/courses/android-basic-kotlin/course

Course code: 22CS351

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

(Common to CSE, IT)

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 develop an understanding of the basic concepts of Artificial Intelligence.

2 : To analyze the nature of various advanced search strategies in AI.

3 : To design and develop model-based techniques to solve classification and clustering problems.

4 : To study the experiential leaning models and designing of adaptative models.

Unit I - Introduction

Introduction to AI, Intelligent Agents.
Solving Problems by Search: Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search.
Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions.
Search in complex environments: Local search and Optimization problems, Local Search in Continuous Spaces, Searching with Non-Deterministic Actions.
Adversarial Search: Optimal decisions in games, Heuristic Alpha-Beta search, Monte Carlo tree search.

Unit II - Knowledge, Reasoning and Planning

Logic Agents: Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving.
First-Order Logic: Syntax and
Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.
Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Unit III - Knowledge Representation

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.
Machine Learning: Attributes, categories of attributes, Estimation of Missing Values Classification, Types of Machine learning
Classification: Basic concepts, Naive Bayes, Decision Tree, Classification Trees, Pruning, Rule Extraction form Tree.

Unit IV - Supervised and Unsupervised Learning

Ensemble Methods: Bagging- Random Forest, Boosting- Gradient Boosting and Ada-boost.
Clustering: Basic Concepts and Methods, Cluster Analysis, Requirements,
Basic clustering methods: Partitioning Methods- k-Means, k-Medoids,
Hierarchical Methods- Agglomerative versus Divisive Hierarchical Clustering.
Distance Measures in algorithmic method- BIRCH, Chameleon
Evaluation of Clustering: Assessing Clustering Tendency, Determining the Number of Clusters, Measuring Clustering Quality.

Unit V - Reinforcement Learning

Introduction to Reinforcement Learning
Elements of RL: Agent, Environment, Reward
Markov Decision Processes (MDPs): Markov Processes and Markov Reward Processes
Introduction to MDPs: States, Actions, Rewards, Policy
Dynamic Programming: Policy Evaluation and Improvement, Value Iteration and Policy Iteration. **Temporal Difference Learning and Q-Learning:** TD Prediction, TD(0) and TD(λ), Q-Learning, Off-policy control using Q-learning.

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

CO 1 : Identify the scope for agent-based engineering solutions using AI based tools
CO 2 : Demonstrate advanced search strategies, perform search space reduction techniques using minmax algorithm
CO 3 : Apply knowledge representation, reasoning to AI-based solutions
CO 4 : Develop multiple-class classifiers and develop recommender systems.
CO 5 : Implement simple Q-Learning algorithm, based on Value iteration, and appreciate actor-critic model

Textbooks:

1. Artificial Intelligence: A Modern Approach, Stuart Russel and Peter Norvig, 4th Edition, Pearson, 2020.
2. Introduction to Machine Learning, Ethem Alpaydin, Francis Bach, 3rd edition Adaptive Computation and Machine Learning series 2014.

References:

1. Machine Learning, Anuradha Srinivasa Raghavan, Vincy Joseph, Paperback, Kindle edition, 2019.
2. Machine Learning using Python, Manaranjan Pradhan U Dinesh Kumar, Paperback, Wiley Publication, 2019.
3. Machine learning, Tom Mitchel, 2012 edition.

Course Code: 22CS352

FULL STACK DEVELOPEMENT

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

Course Objectives:

1 : To enhance code readability, maintainability, and developer productivity by adopting ES6 features and explore version control system.

2 : To gain proficiency and expertise in utilizing NPM (Node Package Manager) and developing RESTful APIs using Express framework.

3 : To explore how NoSQL databases, particularly MongoDB, can be leveraged to design scalable and flexible data solutions.

4 : To Gain proficiency in front-end development using Angular.

5 : To provide the knowledge of building modular and reusable components and create interactive user interfaces for MERN stack applications.

Unit I – Working with ES6 Features

Version Control System: Git, creating local and remote repositories. **Git commands:** init, status, add, commit, remote, push, pull, clone, Git Branching. Using **github.com** for collaborative software development.
ES6 Features extended: variable declaration using var, let, const, Prototypal Inheritance, Classes, Objects, Destructuring, Modules, Symbols, Function Generators, Understanding Callbacks, Closure, Promise, XHR: response, Asynchronous Task in JS, using async / await.

Unit II – Server-side JavaScript – Node JS

Getting started with Node.js: Introduction to Node.js, REPL, NPM. **Node Modules:** os, path, util, events, fs, buffers, streams, http. Building own API and consuming it. (**REST full API**).
Building an Express web application:
Introduction to Express, Installation of Express, create first Express application, application, request, and response objects, Configuring an Express application, rendering views using EJS.

Unit III – Data storage and manipulation

No SQL: Introduction to NoSQL, SQL vs NoSQL, Migrating from SQL to NoSQL database. Different Types of NoSQL databases, CAP Theorem, Sharding.
MongoDB: Introduction to MongoDB, Key features of MongoDB, MongoDB shell, MongoDB databases, MongoDB collections, MongoDB CRUD operations, Real – time database Firebase CRUD operations.
Express-MongoDB connectivity: Connect Express application with MongoDB using mongoose library. Managing user authentication, understanding authorization using Passport OAuth / JWT.
Type Script: Introduction to Type Script, basic types and any type, compiler options, Classes, Interfaces, Generics, Decorators.

Unit IV – Client-side JS Framework Angular

Angular Development: Introduction to Angular, versions, Angular Architecture, Components, Modules, Directives. **Data Binding:** Interpolation, Property Binding, Class Binding, Event Binding, Two – way data binding.
Angular Services: Creating Services, Creating API and Consuming, Dependency Injection.
Routing: Routes, Router Outlet, Router Links. Using HttpClient, Observables, Pipes.
Angular Forms: Template-driven forms and Reactive Forms, passing data from parent to child and passing data between siblings.

Unit V – Client-side JS Library React

Introduction to React JS: Motivation for using React, Key differentiators (Virtual DOM, One-way binding), JSX. **React Components:** Functional Component, Class Component, Render function, Component Lifecycle, State, Props.

Components inter communication: Pass data from parent to child, Pass data from child to parent, fetching data from an API using Axios. **React Routing:** Form Validations, Posting Data, React Router, Building & Deploying React App.

Course Outcomes: At the end of the course, the student will be,

CO 1 : Proficient in leveraging ES6 capabilities to streamline development tasks, improving code efficiency, and solving programming challenges with a modern JavaScript mindset and managing various versions of the product using git.

CO 2 : Able to develop the skills to construct scalable and maintainable web applications by harnessing the asynchronous, event-driven nature of Node.js and the modular architecture provided by Express.

CO 3 : Designing and implementing efficient NoSQL database solutions, understanding when and how to use document-oriented databases like MongoDB to address specific application requirements.

CO 4 : Developing the dynamic and single-page web applications using the MEAN stack and understand how to manage client-side routing, and UI components to deliver a seamless user experience.

CO 5 : Excelling in both server-side and client-side aspects of web application development using MERN stack.

Textbooks:

1. MEAN Web Development, Amos Q. Haviv, Second Edition, Packt Publications, November 2016.
2. "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node", Vasan Subramanian, 2nd Edition, APress, 2017.

References:

1. Learning Node: Moving to the Server-Side, Shelly Powers, 2nd Edition, O'REILLY, 2016.
2. Getting MEAN with Mongo, Express, Angular, and Node, Simon D. Holmes and Clive Harber, Second Edition, Manning Publications, 2019.
3. Node.js, MongoDB and Angular Web Development, Brad Dayley, 2nd Edition, Addison-Wesley Professional, 2017.
4. https://angular-2-training-book.rangle.io.
5. https://www.atlassian.com/git
6. https://www.typescriptlang.org/docs/handbook/basic-types.html
7. https://firebase.google.com

Course Code: 22CS353

CLOUD COMPUTING AND DEVOPS

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 impart concepts in fundamentals of cloud computing, principles, characteristics, and cloud deployment models.

2 : To Learn and understand Virtualization concepts.

3 : To explore various case studies on cloud service providers.

4 : Understand configuration management; continuous integration deployment, delivery and monitoring using DevOps tools such as Git, Jenkins in a practical, hands-on and interactive approach can be explored.

Unit I – Cloud Computing Concepts

Principles of Parallel and Distributed Computing, Introduction to Cloud computing, Cloud computing architecture, Cloud concepts and technologies, Cloud benefits and challenges, Cloud service delivery models – Infrastructure as a Service, Platform as a Service, Software as a Service, Cloud deployment models – public, private, hybrid.

Unit-II - Virtualization and Containers

Introduction, Characteristics of Virtualized Environments, Taxonomy of Virtualization Techniques, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples, Contemporary Virtualization Techniques, Containers, Container Orchestration, Docker and Kubernetes, Terraform.

Unit III – Cloud Case Studies

Case studies of Google Cloud Platform, Cloud Toolkit , Google App Engine, Amazon Web Services, Microsoft Windows Azure .

Unit IV – Introduction to CI / CD

Introduction to DevOps: What Is DevOps? DevOps Importance and Benefits, DevOps Principles and Practices, 7"Cs of DevOps Life Cycle for Business Agility, Continuous Planning, Continuous Development, Continuous Integration, Continuous Deployment, Continuous Testing, Continuous Delivery and Monitoring, Continuous Feedback, DevOps and Continuous Testing, Steps to be followed to choose the right DevOps Tools, Selecting the Right Tools, Challenges with DevOps Implementation, Must Do Things for DevOps.

Unit V – Devops Tool Suites

Tool Suites: Atlassian Tools, Phabricator
Orchestration: Jenkins-Features, Example of Reference Architecture, Microsoft TFS-Features, Example of Reference Architecture, TeamCity- Features, Example of Reference Architecture, Ansible Features, Example of Reference Architecture, Bamboo- Features, Example of Reference Architecture.
Source Code Management and Quality: Bitbucket- Features, Example of Reference Architecture, Crucible- Features, Example of Reference Architecture.

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

CO 1: Understand the Cloud computing fundamental concepts.

CO 2: Learn about Virtualization, and contemporary virtualization concepts like Containers, and Dockers.

CO 3: Explore Cloud Computing case studies like Amazon Cloud, Google App Engine, and Microsoft Azure.

CO 4: Analyze need for continuous integration and appreciate continuous deployment in Industrial Scenario

CO 5: Implement GitHub and Jenkins for configuration management and continuous integration.

Textbooks:

1. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi, Mc Graw Hill, 2017.

References:

1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O’Reilly, SPD, rp 2011.
2. DevOps Tools from Practitioner's Viewpoint, Deepak Gaikwad, Viral Thakkar, Wiley, 2019.
3. Essentials of cloud Computing: K. Chandrasekhran, CRC press, 2014.
4. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Course code: 22CS354/22CS309

AUTOMATA THEORY AND COMPILER DESIGN

(Common to CSE, CSE(AI&ML), CSE(CS), CSE(DS))

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 the concepts of regular languages, finite automata, regular expressions, and context free grammar.
2 : To make the students understand and implement top-down and bottom-up parsers.
3 : To make the students understand the intermediate code forms, type checking.
4 : To acquire knowledge on storage allocation strategies, symbol table management and code generation algorithms.

Unit I – Introduction to Automata

Languages, definitions, Regular Expressions, Regular Grammars, Acceptance of Strings and Languages, Finite Automaton Model, DFA, NFA, conversion of NFA to DFA, Conversion of Regular Expression to NFA, Chomsky hierarchy of Languages.

Unit II– Lexical Analysis and Top-down Parsing

Phases of compilation overview, Pass, Phase, Interpretation, Bootstrapping.
Context free grammars, Top-down Parsing: Parse Trees, Ambiguous Grammars, Backtracking, LL (1), Recursive Descent parsing, Predictive parsing, pre-processing steps for predictive processing.

Unit III – Bottom-Up Parsing and Syntax Directed Translation

Bottom-up parsing and handle pruning, LR (k) grammar parsing, LALR (k) grammars, Error Recovery in parsing, parsing ambiguous grammars, YACC parser generator.
Syntax Directed Translation, Attribute Grammars, Evaluation order for SDDs, Syntax Directed Translation schemas, Intermediate source program forms - AST, polish notation and 3 address code, DAG, Types and declarations, Type Checking, Equivalence of type expressions.

Unit IV – Code Optimization

Symbol table format, organization, Block structured languages, hashing, Block structure and non- block structure storage allocation: static, runtime and heap allocation for arrays, strings, and activation records.

Consideration for optimization, Scope of optimization, DAG representation, Basic blocks, partitioning into basic blocks, flow graphs, Compile Time Evaluation, Common Subexpression elimination, dead code elimination, Strength Reduction, Code Movement, Loop Invariant Method, Loop Fusion, Loop Unrolling, Induction Variables and Reduction in Strength.

Unit V – Code Generation

Absolute Code, Assembler Code, Register and Address Descriptors, Implementing Global Register Allocation, Usage Counts, Using DAG for register allocation, Simple Code generation Algorithm, Generic Code generation Algorithm.

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

CO 1: Covert NFA to DFA and regular expression to DFA.
CO 2: Design top-down and bottom-up parsers.
CO 3: Understand the concepts of type checking and intermediate code generation forms.
CO 4: Understand the concepts of storage allocation strategies, symbol table management and hashing
CO 5: Use DAG for generating assembly language code and able to generate relocatable machine code.

Textbooks:

1. Introduction to Automata Theory Languages and Computation, Hopcroft H.E. and Ullman J.D., Pearson Education, 2009.
2. Principles of Compiler Design, A.V Aho and J D Ullman, Pearson Education, 2002.

References:

1. Compiler Construction: Principles And Practice, Kenneth C. Louden, Thomson/D elmar Cengage Learning, 2006.
2. Lex & yacc, Doug Brown, John Levine and Tony Mason, 2nd Edition, O’reilly Media, 1992.
3. Engineering a compiler, Keith Cooper and Linda Torczon, 2nd Edition, Morgan Kaufmann,2011.
4. Modern Compiler Construction in C, Andrew W. Appel, Cambridge University Press, 2004

Course code: 22CS355

CRYPTOGRAPHY AND ESSENTIALS OF NETWORK SECURITY

(Professional Elective - II)

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 : Understand the various Symmetric and Asymmetric Cryptographic algorithms.

2 : Appreciate various Key Management and Distribution along with Authentication schemes.

3 : Understand the different protocols used for Network security and System security.

4 : Introduce the different aspects of Application-level security.

Unit – I- Security and Cryptography concepts

Security Concepts: Introduction, Security Attacks, Security Services, Security Mechanisms, A model for Network Security.
Cryptography Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric keycryptography, steganography.
Symmetric key Ciphers: Symmetric Encryption principles, DES, AES, Block cipher modes of operation, Stream ciphers, RC4.

Unit – II- Introduction to Modular Arithmetic

Asymmetric key Ciphers: Public key cryptography principles, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange.
Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

Unit – III-Cryptographic Hash Functions

Approaches to Message Authentication, Secure Hash Algorithm (SHA-512).
Message Authentication Codes: Authentication requirements, HMAC, Digital signatures, Elgamal Digital Signature Scheme.
IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange.
.

Unit – IV-Transport-level Security

Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH).
E-Mail Security: Pretty Good Privacy, S/MIME

Unit – V- Malicious Software

Types of malicious software, Types of attacks, Countermeasures, DDoS attacks.
Intruders: Intrusion detection and Password Management.
Firewalls: Characteristics, Access Policy, Types of Firewalls, Firewall Basing, Location and Configurations

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

CO 1 : Understand basic Cryptographic concepts and Symmetric Key algorithms.

CO 2 : Understand and analyze Public-Key Cryptography along with Key Management and Distribution

CO 3 : Analyze and design Hash and MAC algorithm and IP security.

CO 4 : Understand the Transport level security and Web security.

CO 5 : Understand the Intruders, Viruses and Firewalls.

Textbooks:

1. Cryptography and Network Security - Principles and Practice, William Stallings, PearsonEducation, 7th Edition, 2017
2. Network Security Essentials: Applications and Standards: William Stallings, Pearson Education, 6th Edition, 2018
3. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition, 2017

References:

1. Cryptography and Network Security, C K Shyamala, N Harini, Dr T R Padmanabhan, WileyIndia, 1st Edition, 2006
2. Cryptography and Network Security: Forouzan Mukhopadhyay, Mc Graw Hill, 3rd Edition,2015.
3. Information Security, Principles, and Practice: Mark Stamp, Wiley India, 2011.
4. Principles of Computer Security: WM. Arthur Conklin, Greg White, TMH,2016.
5. Introduction to Network Security: CENGAGE Learning, Neal Krawetz, 2007.
6. Network Security and Cryptography: CENGAGE Learning, Bernard Menezes, 2010.

Course Code: 22CS356

ADHOC AND SENSOR NETWORKS

(Professional Elective - II)

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 : Understand MANETs, routing approaches, and protocols in wireless networks.
2 : Analyze data transmission challenges, TCP protocol, and solutions in MANETs.
3 : Explore Wireless Sensors, WSN classification, MAC layer, and adaptability.
4 : Study security in Ad Hoc Networks, including key management.
5 : Familiarize with TinyOS, languages, and node-level simulation tools.

Unit I - Introduction to Ad hoc Wireless Networks

Introduction to Ad hoc Wireless Networks: Characteristics of MANETs, Applications of MANETs, Challenges.
Routing in MANETs: Topology-based versus Position-based approaches, Topology based routing protocols, Position based routing, Other Routing Protocols.

Unit II - Data Transmission in MANETs

Data Transmission in MANETs: The Broadcast Storm, Multicasting, Geo-casting
TCP over Ad hoc Networks: TCP Protocol overview, TCP and MANETs, Solutions for TCP over ad hoc networks

Unit III - Basics of Wireless Sensors and Applications

Basics of Wireless Sensors and Applications: Introduction, The Mica Mote, Sensing and Communication Range, Design issues, Energy consumption, Clustering of Sensors, Applications of WSNs.
Data Retrieval in Sensor Networks: Classification of WSNs, MAC layer, Routing layer, High level application layer support, Adapting to the inherent dynamic nature of WSNs.

Unit IV – Sensor Operating System

Operating System: TinyOS, Imperative Language: nesC Dataflow style language: T1nyGALS, Node- Level Simulators: ns-2 and its sensor network extension, TOSSIM.

Unit V – Security in MANET

Security: Security in Ad hoc Wireless Networks, Key Management, Secure Routing, Cooperation in MANETs, Intrusion Detection Systems.
Sensor Network Platforms and Tools: Sensor Network Hardware, Sensor Network Programming Challenges, Node-Level Software Platforms.

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

CO 1: Analyze MANETs, choose appropriate routing protocols for wireless networks.
CO 2: Address data transmission challenges, optimize TCP in dynamic environments.
CO 3: Design efficient Wireless Sensor Networks, considering various application scenarios.
CO 4: Develop applications using Tiny OS, simulate networks for analysis.
CO 5: Implement secure Ad Hoc Networks, manage keys, detect intrusions.

Textbooks:

1. Ad Hoc and Sensor Networks: Theory and Applications, Carlos De Morais Cordeiro and Dharma Prakash Agrawal, World Scientific Publications /Cambridge University Press, March 2006.
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao and Leonidas Guibas , Elsevier Science imprint, Morgan Kauffman Publishers, Reprint 2009.

References:

1. Ad hoc Wireless Networks: Architectures and Protocols, C. Siva Ram Murthy and B.S. Murthy, Pearson Education, 2004.
2. Wireless Sensor Networks: Principles and Practice, Fei Hu, Xiaojun Cao, Auerbach / CRC Press, Taylor & Francis Group, 2010.
3. Wireless Ad hoc Mobile Wireless Networks: Principles, Protocols and Applications, Subir Kumar Sarkar et al. Auerbach Publications, Taylor & Francis Group, 2008.
4. Ad hoc Networking, Charles E. Perkins, Pearson Education, 2001.
5. Wireless Ad hoc Networking, Shih-Liri Wu and Yu-Chee Tseng, Auerbach Publications, Taylor & Francis Group, 2007

Course Code: 22CS357

INTERNET OF THINGS

(Professional Elective - II)

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 impart necessary and practical knowledge of components of IoT.

2 : To demonstrate the various python packages, interfacing Raspberry Pi was used for application development.

3 : To Learn and understand the IoT and M2M fundamental concepts, and system management.

4 : To Learn and Understand the IoT and cloud computing Architecture, service models, Web application framework and APIs.

5 : To Implement the Domain specific IoT applications in Realtime environment like Home, city, energy, antiagriculture, health, Industry and Lifestyle.

Unit I –Introduction to Internet of Things

Introduction - Definition and Characteristics of IoT. Physical Design of IoT – IoT Protocols. Logical Design of IoT -IoT Communication Models, IoT Communication APIs. IoT Enabling Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems. IoT Levels and Deployment Templates.

Unit II - Python packages and IoT Interfaces

JSON, XML, HTTPLib, URLLib, SMTPLib, RPi.GPIO.
IoT Physical Devices and Endpoints - Introduction to Raspberry PI, Interfaces (serial, SPI, I2C). Programming Raspberry PI with Python - Controlling LED, interfacing an LED and Switch, Interfacing a Light Sensor with Raspberry Pi.

Unit III – IoT and M2M

Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT. IoT System Management with NETCONF-YANG - SNMP, NETCONF, YANG, IoT System Management with NETCONF-YANG, NETOPEER.

Unit IV – IoT Physical Servers and Cloud Offerings

Cloud Computing – Definition, Characteristics, Architecture, Service Models and Deployment Models, Virtualization Concepts. Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Cloud for IoT, Python Web Application Framework, Designing a RESTful Web API.

Unit V – Domain Specific IoTs

Introduction - Home, City, Environment, Energy Systems, Retail, Logistics, Agriculture, Industry, Health, and Lifestyle.

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

CO 1: Understand the characteristics, protocols and communication models required for logical design of IoT.

CO 2: Realize the hardware platforms for implementing and interfacing the IoT based board with different peripheral devices and serial communication devices.

CO 3: Gain knowledge on protocol stacks for IoT and M2M networks and configurations.

CO 4: Integrate devices and develop an application that can communicate through IoT Cloud.

CO 5: Implement various case studies and applications in IoT design.

Textbooks:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015.
2. Learning Internet of Things, Peter Waher, Packt publisher, 2015.
3. Getting Started with Raspberry Pi ,Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014.

References:

1. Cloud Computing: Principles and Paradigms, Raj Kumar Buyya, James Broberg and Andrzej M Goscinski, Wiley, 2013
2. Getting started with sensors: Measure the world with Electronics, Arduino, and Raspberry, Kimmokarvinen and tero Karvenien, First Edition, Shroff/O"Reilly, 2014.
3. Getting started with Raspberry Pi, Richardson Matt, Shroff Publishers & Distributers Private Limited,2012

Course code: 22CS358

DISTRIBUTED SYSTEMS

(Professional Elective - II)

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

Course Objectives:

1 : To introduce the characteristics of distributed systems and inter-process communication.

2 : To enrich the knowledge of processes, threads, and operating system architecture.

3 : To acquire knowledge on peer-to-peer systems and their applications.

4 : To make the students understand the concepts of transactions and concurrency control in distributed systems.

5 : To equip students with principles and knowledge of transactions with replicated data.

Unit I - Characterization of Distributed Systems

Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models -Introduction, Architectural and Fundamental models, Networking and Internetworking, Inter-process Communication, Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

Unit II - Operating System Support

Introduction, OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture, Distributed File Systems-Introduction, File Service architecture.

Unit III Peer to Peer Systems

Introduction, Napster and its legacy, Peer to Peer middleware, Routing overlays,Overlay case studies- Pastry, Tapestry, Application case studies - Squirrel, Ocean Store.
Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physicalclocks, logical time and logical clocks, global states, distributed debugging.
Coordination and Agreement-Introduction, Distributed mutual exclusion, Elections, Multicast communication, consensus, and related problems.

Unit IV - Transactions and Concurrency Control

Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering. Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

Unit V - Replication

Introduction, System model and group communication, Fault-tolerant services,Transactions with replicated data.
Distributed shared memory, Design and Implementation issues, and Consistency models.

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

CO 1: Understand the characteristics of distributed systems and inter-process communication.

CO 2: Understand the concepts of processes, threads, and operating system architecture.

CO 3: Design a peer-to-peer systems and their applications.

CO 4: Understand the concepts of transactions and concurrency control in distributed systems.

CO 5: Use replicated data in transactions in distributed systems.

Textbooks:

1. Distributed Systems Concepts and Design, G Coulouris, J Dollimore and T Kindberg , FourthEdition, Pearson Education, 2008.
2. Distributed Systems, S.Ghosh, Chapman & Hall/CRC, Taylor & Francis Group, 2010.

References:

1. Distributed Systems – Principles and Paradigms, A.S. Tanenbaum and M.V. Steen, PearsonEducation, 2016.
2. Distributed Computing, Principles, Algorithms and Systems, Ajay D. Kshemakalyani andMukesh Singhal, Cambridge, rp 2010.

Course code: 22CS359

ARTIFICIAL NEURAL NETWORKS AND GRAPHICAL MODELS

(Professional Elective - II)

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

Course Objectives:

1 : To develop an understanding of the basic concepts of Artificial Intelligence.

2 : To gain the rationale behind the capabilities of different ANN architectures.

3 : To impart knowledge representation, non-linear boundaries.

4 : To derive value from Random Fields

Unit I – Multilayer Perceptron and Back Propagation

Introduction: A Neural Network, Human Brain, Models of a Neuron, Neural Networks viewed as Directed Graphs.
Learning Process: Error Correction Learning, Memory Based Learning, Hebbian Learning, Competitive, Boltzmann Learning, Credit Assignment Problem.
Single Layer Perceptron: Adaptive Filtering Problem, Least Mean Square Algorithm, Perceptron –Convergence.
Multi-Layer Perceptron: Back Propagation Algorithm, XOR Problem, Heuristics to improve the performance. Regularization, Weight Initialization techniques, Momentum based training.

Unit II – Unsupervised Learning and Sequence Learning

Associative Memory Networks: Training algorithms for pattern association, BAM and ART1.
Self-Organization Maps (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification.
Sequence Learning: Recurrent Network Paradigm, BPTT Algorithm, Hopfield Networks.

Unit III – Graphical Models

Bayesian Networks: Definition, Example: Polynomial regression, Generative models, Discrete variables, Linear-Gaussian models, Conditional Independence, Three example graphs, D-separation.
Markov Random Fields: Conditional independence properties, Factorization properties, Illustration: Image de-noising, Relation to directed graphs, Inference in Graphical Models, Inference on a chain, Trees, Factor graphs, The sum-product algorithm, The max-sum algorithm, Exact inference in general graphs, Loopy belief propagation, Learning the graph structure.

Unit IV – Mixture Models and EM

Mixtures of Gaussians, Maximum likelihood, EM for Gaussian mixtures, EM for Bayesian linear regression, EM Algorithm in General.
Approximate Inference: Variational Inference, Factorized distributions, Properties of factorized approximations, Example: The univariate Gaussian, Model comparison, Illustration: Variational
Mixture of Gaussians: Variational distribution, Variational lower bound, Predictive density, Determining the number of components, Induced factorizations

Unit V – Sampling Methods

Basic Sampling Algorithms, standard distributions, Rejection sampling, Adaptive rejection sampling, Sampling-importance-resampling, Sampling and the EM algorithm

Markov Chain Monte Carlo: Markov chains, Metropolis-Hasting's algorithm, Gibbs Sampling, Slice Sampling, Hybrid Monte Carlo Algorithm, Dynamical systems Hybrid Monte Carlo, Estimating the Partition Function.

Course Outcomes: After completion of course, students will be able to

CO 1 : Implement Backpropagation algorithm and solve non-liner classification problem
CO 2 : Solve semi-supervised learning-based SOM
CO 3 : Address real-world image restoration problem using Belief Propagation algorithm
CO 4 : Appreciate Variational principle based MLE approximation
CO 5 : Gain proficiency in different sampling methods

Textbooks:

1. Neural Networks a Comprehensive Foundations, Simon Haykin, 2nd Edition, Pearson, 2009.
2. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006.

References:

1. Artificial Intelligence: Structures and Strategies for complex problem solving, G. Luger, 4thEdition, Pearson Education, 2002.
2. Artificial Intelligence: A new Synthesis, J. Nilsson, Elsevier Publishers, 1998.
3. Probabilistic Graphical Models, Koller, and N. Friedman, MIT Press, 2009.

Course Code: 22CS381

INTERNET OF THINGS AND CLOUD COMPUTING LAB

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

List of Experiments: IOT

Task 1:

Installing Node-RED on a Raspberry Pi through Remote Login.

Task 2:

Create a simple Node-RED flow that takes input from an inject node and displays output in a debug node. Add a function node to modify the payload in the flow.

Task 3:

Integrate an MQTT node into a Node-RED flow and subscribe to a topic.

Task 4:

Implement an HTTP request node to interact with an external API. (Twitter, WhatsApp)

Task 5:

Build a basic IoT dashboard using the Node-RED dashboard nodes.

Task 6:

Include widgets for displaying sensor data and control buttons.

Task 7:

Configure security settings for Node-RED, including user authentication.

Task 8:

Implement SSL/TLS for secure communication in a Node-RED instance.

List of Experiments : Cloud Computing

Task 1:

Create and Manage Cloud Resources

a) Tour of Google Cloud
b) Creating a Virtual Machine
c) Getting Started with Cloud Shell and g cloud

Task 2:

Kubernetes Engine:

a) Set Up Network and HTTP Load Balancers
b) Create and Manage Cloud Resources: Challenge Lab

Task 3:

Perform Foundational Infrastructure Tasks

a) Cloud Storage: Qwik Start - Cloud Console
b) IAM in AWS and setup
c) Cloud Functions
d) Cloud networking

Task 4:

Set Up and Configure a Cloud Environment
a) Cloud IAM: Qwik Start
b) Introduction to SQL for Big Query and Cloud SQL
c) Database Tasks in Cloud
d) Cloud Monitoring: Qwik Start
e) Managing Deployments Using Kubernetes Engine
f) Set Up and Configure a Cloud Environment in Google Cloud: Challenge Lab

Task 5:

1. Introduction to Amazon EC2
2. Introduction to Amazon Simple Storage Service (S3)

Task 6:

1. Introduction to Amazon Relational Database Service (RDS) - SQL Server)
2. AWS Identity and Access Management (IAM) Task

Task 7:

1. Management of Amazon Elastic Container Service
2. Hosting a static website and Deploying a web application on AWS

Task 8:

a. Continuously Querying top 10 songs in the song-list or chart with Kubernetes
b. Key Parameter Indicators visualization for Airline services using Kubernetes

Task 9:

a. Managing resources using Terraform
b. creating and running containers.

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

CO 1 : Implement framing, error detection techniques of data link layer.
CO 2 : Design and implement different routing algorithms in network layer.
CO 3 : Design and Implement security algorithms in application layer.
CO 4 : Understand the fundamentals of creating and managing cloud
CO 5 : Identify the significance of Kubernetes engine and process of tasks execution

References:

1. Computer Networks, Andrew S Tanenbaum, 5th Edition, Pearson Education, 2011.
2. Data Communications and Networking, Behrouz A. Forouzan, 5th Edition, TMH, 2009.
3. Network Security Essentials: Applications and Standards, William Stallings, 4th Edition Pearson Education, 2011
4. Cryptography and Network Security, B.A. Forouzan and D. Mukhopadhyay, TMH, 2ndEdition, TMH. 2010.

Course Code: 22CS382

FULL STACK DEVELOPMENT LAB

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

Course Objectives:

1 : To enhance code readability, maintainability, and developer productivity by adopting ES6 features and exploring version control system.

2 : To gain proficiency and expertise in utilizing NPM (Node Package Manager) and developing RESTful APIs using Express framework.

3 : To explore how NoSQL databases, particularly MongoDB, can be leveraged to design scalable and flexible data solutions.

4 : To Gain proficiency in front-end development using Angular.

5 : To provide the knowledge of building modular and reusable components and create

Task 1: Managing the versions of the product using Version Control System (GIT)

1. Create a git repository local add and commit a simple web application consisting of 5 pages.
2. Create a remote repository in github.com. Push the above local repository to the github.com. Explore the push, pull and fetch options with remote repository.
3. Clone a remote repository into local directory, modify the implementation and push the updated changes back to remote repository.
4. Create branch and manage the work distributions. Merge all the branches and commit the changes.
5. Publish the application using GitHub pages.

Task 2: Implementing the advanced features of JavaScript

1. Working with Prototypal Inheritance and Classes.
2. Working with Object and Array Destructuring.
3. Working with Modules.
4. Working with Function Generators and Symbols.
5. Working with Closure.

Task 3: Asynchronous Execution of JavaScript

1. Working with higher order function in JavaScript.
2. Using Callback and creating a Callback Hell situation to understand the drawbacks.
3. Working with XHR: response.
4. Dealing with the Callback Hell situation using Promise. Exploring the different ways of creating and using promise in executing the asynchronous task.
5. Dealing with Promise chaining and async / await.

Task 4: Fetching server state using JavaScript (fetch)

1. Use fetch function to access remote data using the given api and display the data in the form of a table.
2. Use fetch function to read the weather details from openweathermap.org and display the details like city, min-temp, max-temp, humidity on the webpage for a given city.
3. From the same website read the weather forecast details for a given city and display the details like date – temperature in a table.
4. Plot a bar chart for the above implementation using date and temperature along X and Y axis respectively. Use ChartJS library.

Task 5: Node JS

1. Create custom / local modules and export them using various module patterns.
2. Explore the functionality of os, path, util and events modules.
3. Use the fs module for creating directories and files of different formats.
4. Write script to read and write the streaming data using readable and writable streams.

Task 6: Working with http

1. Create a http server listening request at port 3000. Process the request to provide different type of resources as response. (HTML, TEXT, JSON, etc.).
2. Create express server listening request at port 3000. Add different endpoints to provide access to the resources.

Task 7: Working with Express

1. Create a custom API for Users data and add different endpoints in express server to perform CRUD operations on the API. Test the endpoints using POSTMAN.
2. Use EJS view-engine to display the dynamic response. Display the data read from REST API in the form of a table in EJS.

Task 8: Working with model (MongoDB & Firebase)

1. Create a database in Mongo DB. Create Users collection and documents to the User collection. Perform all DB operations (CREATE, READ, UPDATE and DELETE) on the User collection.
2. Create a real time database in firebase for the student management system and explore the features of Firebase Real Time Database. Perform CRUD operations on the Real Time Database.

Task 9: Working with Express & Mongo DB

1. Create express server that has endpoints connecting to Users collection present in Mongo DB database using mongoose library and perform CRUD operation on that.

Task 10: Authentication and Authorization

1. Create express server that has authorized endpoint using JWT (JSON Web Token) library.
2. Create express server that connects to Mongo DB database to authenticate the user and generate the authorized token to access the protected endpoints.

Task 11: MEAN Stack Development

1. Create a user profile management system where users can update their profiles, including details like name, email, phone. Use Angular's data binding to ensure that changes made by users are instantly reflected in the UI.
2. Develop an angular application that interacts with the backend API and executes CRUD operations on it.

Task 12: Angular Routing

1. Develop angular application consisting of App, Home, About, Contact, Profile, Login and Register Components. Add a Navigation bar and navigate to the respective component using angular routing.
2. Develop a Single Page Application in Angular for User Management System that interacts with the backend database created in Task 8. Use Services and Http-Client to access the express endpoints of Task 9.

Task 13: MERN Stack Development

1. Create react functional and class components. Implement the lifecycle methods of react component.
2. Develop react application with App, Home, About and Contact components. Implement the use of react props and state in these components.

Task 14: React Routing

1. Develop a react application that demonstrates the routing feature to navigate across different components of react and pass the data in between the components.
2. Develop a SPA in react for User Management System that interacts with the backend API using Axios and perform CRUD operations on that.

Task 15: Single Page Application in Angular / React

A TODO application serves as a simple yet powerful tool to help individuals and teams organize their tasks, manage priorities, and enhance productivity. TODO applications provide a structured and efficient way for individuals and teams to manage tasks, prioritize work, and achieve their goals. Develop a Single Page TODO Application in Angular React to manage the daily tasks with the following features:

i. **Task Creation:** Allow users to create new tasks with a title, description, due date, and priority level. Provide a straightforward interface for entering task details.
ii. **Task Listing:** Display a list of all tasks with essential details. Tasks can be organized based on different criteria such as due date, priority, or completion status.
iii. **Task Editing and Updating:** Enable users to edit task details, including the ability to modify the title, description, due date, and priority. Changes should be reflected in real-time.
iv. **Task Deletion:** Provide the option to delete tasks that are no longer relevant or completed. Include a confirmation prompt to prevent accidental deletions.
v. **Task Completion:** Allow users to mark tasks as completed or mark them with a specific status. Completed tasks may be moved to a separate section or visually differentiated.
vi. **User Authentication and Authorization:** Implement user accounts with authentication to ensure data privacy. Differentiate between users and provide appropriate authorization levels.
vii. **Data Persistence:** Ensure that tasks are persistently stored, so users can access their TODO lists even after closing and reopening the application.

(NOTE: Task 1 – Task 14 are mandatory to complete in the labs)

Course Outcomes: At the end of the course, the student will be,

CO 1 : Proficient in leveraging ES6 capabilities to streamline development tasks, improving code efficiency, and solving programming challenges with a modern JavaScript mindset and managing various versions of the product using git

CO 2 : Able to develop the skills to construct scalable and maintainable web applications by harnessing the asynchronous, event-driven nature of Node.js and the modular architecture provided by Express

CO 3 : Designing and implementing efficient NoSQL database solutions, understanding when and how to use document-oriented databases like MongoDB to address specific application requirements.

CO 4 : Developing the dynamic and single-page web applications using the MEAN stack and understand how to manage client-side routing, and UI components to deliver a seamless user experience

CO 5 : Excelling in both server-side and client-side aspects of web application development using MERN stack

References:

1. MEAN Web Development, Amos Q. Haviv, Second Edition, Packt Publications, November 2016.
2. "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node", Vasan Subramanian, 2nd Edition, APress, 2017.
3. Learning Node: Moving to the Server-Side, Shelly Powers, 2nd Edition, O'REILLY, 2016.
4. "Getting MEAN with Mongo, Express, Angular, and Node", Simon D. Holmes and Clive Harber, Second Edition, Manning Publications, 2019.
5. "Node.js, MongoDB and Angular Web Development", Brad Dayley, 2nd Edition, Addison-Wesley Professional, 2017.
6. https://angular-2-training-book.rangle.io.
7. https://www.atlassian.com/git
8. https://www.typescriptlang.org/docs/handbook/basic-types.html
9. https://firebase.google.com

Course Code: 22HS381/22HS331

ADVANCED ENGLISH COMMUNICATION & SOFT SKILLS LAB

(Common to All branches)

Instruction	:	2 Periods/week	Continuous Internal Evaluation	:	40 marks
Tutorial	:	-	Semester End Examination	:	60 marks
Credits	:	1	Semester End Exam Duration	:	2 hrs

Course Objectives:

1 : To equip students with the requisite communication skills for real-time environment.

2 : To prepare students for persuasive conversations in the professional sphere.

3 : To integrate time management and decision-making skills for better performance.

4 : To modify communication to suit diverse cultures.

5 : To sensitize students to handle emotions at workplace.

Unit I - Soft Skills & Interpersonal communication

a) Effective Communication: types of communication-verbal & non-verbal, importance of communication, 7 Cs of communication, barriers to effective communication, communication according context, strategies for improving communication.

b) Intrapersonal & Interpersonal Skills: definition, how to start a conversation, self-introduction, self-concept, signs of high & low self-esteem, self-exploration, SWOT analysis.

Unit II - Oral Communication

a) Group Discussion- significance of GD, types of GD, opening strategies, roles of participants, evaluation parameters, dos and don’ts, types of topics, features of GD, mock GD

b) Negotiation Skills- definition, process, outcome of negotiation, skills required, strategies of communication for negotiation.

c) Book-discussion- genres of book, importance of book-discussion, purpose of book-discussion, process of book-discussion, critical and analytical skills.

Unit III - Employability Skills

a) Team Dynamics- difference between team and group, types of teams, concepts related to team building, process of team building, roles of team player and qualities.

b) Time Management- concept of time-management, time logs, timewasters, time quadrant, priority list, tips of time management.

c) Decision-making & Problem Solving- strategies of decision-making, techniques of decision-making, problem-solving process.

Unit IV - Cross-cultural Communication

a) Pluralism: Introduction to Cross-cultural Management, Communicating across Cultures, high-low context culture, negotiating across Cultures, Motivation and Leadership across Cultures, Managing Global Teams, Global Manager

b) Diversity & adaptability- race, gender, class, caste, religion.

Unit V - Emotional Intelligence

Concept of Emotional Intelligence: Intrapersonal Awareness, Intrapersonal Management, Conflict Management and Leadership, Anger Management

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

CO 1 : gain proficiency in communication skills.
CO 2 : emerge as rational speakers.
CO 3 : efficiently manage their professional career.
CO 4 : acclimatize to diverse cultures.
CO 5 : be empowered with skills required for self-management.

References:

1. Communication and Soft Skill development, Ashwini Deshpande, 1st Edition, Career Publications, 2017.
2. Effective Communication and Soft Skills, Nitin Bhatnagar and Mamta, Pearson, 2011.
3. Soft Skills & Life Skills, The Dynamics of Success, Nishitesh and Dr. Bhasker Reddi, BSC Publishers & Distributors, 2012.
4. Guide to Cross-Cultural Communications, Reynolds, Valentine and Verma, Pearson, 2010.
5. Emotional Intelligence: A Comprehensive Self Help Guide to Developing EQ, Managing Anger, and Improving your Relationships, Christopher Rance, Ingram Publishing 2019.
6. Unearthing your Emotional Intelligence, Deepa R, Notion Press, Paperback, 2020.

Course Code: 22HS352/22HS302/22HS253

INTELLECTUAL PROPERTY RIGHTS

(Common to CE, CSE, EIE, ECE, IT, ME, EEE and CSE (AI&ML))

Instruction	:	3 Periods/week	Continuous Internal Evaluation	:	100 Marks
Tutorial	:	-	Semester End Examination	:	-
Credits	:	0	Semester End Exam Duration	:	-

Course Objectives:

1 : To impart basic concepts in IPR
2 : To understand the various aspects of Trade Marks.
3 : To create awareness on Law of Patents and Copyrights.
4 : To highlight relevance of Trade Secrets in any Trade/Business
5 : To get elementary knowledge of International IPRs and New developments in IPR.

Unit I - Introduction to Intellectual property

Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

Unit II - Trade Marks

Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

Unit III - Law of copyrights

Fundamental of copy right law, originality of material, rights of reproduction, rights to perform the work publicly, copy right ownership issues, copy right registration, notice of copy right, international copy right law.
Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

Unit IV- Trade Secrets

Trade secrete law, determination of trade secrete status, liability for misappropriations of trade secrets, protection for submission, trade secrete litigation.
Unfair competition: Misappropriation right of publicity, false advertising.

Unit V- New development in intellectual property

New developments in trade mark law; copy right law, patent law, intellectual property audits. International overview on intellectual property, international – trade mark law, copy right law, international patent law, and international development in trade secrets law.

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

CO 1 : understand concepts of intellectual property rights.
CO 2 : evaluate trade mark
CO 3 : file for a patent
CO 4 : analyze the fairness in a competition
CO 5 : understand laws related to intellectual property rights.

Textbooks:

1. Intellectual Property Right, Deborah. E. Bouchoux, 4th Edition, Cengage learning, 2012.
2. Intellectual Property right – Unleashing the knowledge economy, Prabuddha Ganguli, 1st Edition, Tata McGraw Hill Publishing company ltd, 2017.

References:

1. Intellectual Property Patents, Trademarks and Copyrights, Richard Stim, 2nd Edition, Cengage learning, 2012.
2. Intellectual Property Rights under WTO, T. Ramappa, S. Chand, 2008.

Course Code: 22CS401

LINUX PROGRAMMING

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 develop the skills necessary for systems programming.

2: To model asynchronous event handling.

3: To establish efficient communication between two asynchronous processes.

4: To design various client & server communication models

Unit I – File System and Directory Structure

Files: Files concept, File System Structure, I-nodes, File Attributes, File types, Library functions, the standard I/O and formatted I/O in C, stream errors, kernel support for files, System calls, file descriptors, low level file access- File structure related system calls(File APIs), file and record locking, file and directory management – Directory file APIs, Symbolic links & hard links.

Unit II - Process and Signals

Process: Process concept, Kernel support for process, process attributes, process control - process creation, waiting for a process, process termination, zombie process, orphan process APIs.
Signals: Introduction to signals, Signal generation and handling, Kernel support for signal, Signal function, unreliable signal, reliable signal, kill, raise, alarm, pause, abort, sleep functions.
Inter Process Communication: Introduction to IPC, Pipes, FIFOs.

Unit III -IPC, Message Queues, Semaphores, Shared Memory and Socket Programming

Message Queues – Kernel support for messages, Unix system V APIs for messages, client/server example.
Semaphores: Kernel support for semaphores, Unix system V APIs for semaphores.
Shared Memory: Kernel support for shared memory, Unix system V APIs for shared memory, semaphore and shared memory example.
Sockets: Introduction to Sockets, Socket Addresses Structures, Byte ordering and manipulation functions, Socket related system calls for TCP sockets- Socket, connect, bind, listen, fork, exec, and close , Implementation of concurrent server, TCP Client Server programs, Normal startup, terminate and signal handling server process termination. TCP Client-Server Program and UNIX domain Sockets.

Unit IV - Socket Programming, Multithreaded Programming

Socket Options, Server and service models, Super Server, Elementary UDP Sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP.
Multithreaded Programming: Differences between threads and processes, Thread structure and uses. Threads and Lightweight Processes, POSIX Thread APIs, Creating Threads, Thread Attributes, Thread Synchronization with Condition Variables and with Mutexes, Example programs.

Unit V –Advanced I/O

I/O Multiplexing and Socket options: I/O Models, Select Function, poll function.
Record Locking, Readn and Writen functions, Scatter and Gather IO, Memory Mapped IO. Asynchronous IO and Async Options. Remote login overview and RPC Transparency issues. FTP server configuration.

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

CO 1 : Make use of well-defined Korn shell utilities and develop menu driven Text processing Application.
CO 2 : Appreciate process abstraction and asynchronous event handling using signals.
CO 3 : Implement IPC Mechanisms, Messages Queues and synchronize the access patterns as a shared memory.
CO 4 : Design concurrent server programs based on various design alternatives.

CO 5 : Implement I/O multiplexing mechanisms.

Textbooks:

1. Advanced Programming in the UNIX Environment, W Richard Stevens and Stephen A Rago, 3rd Edition, Addison Wesley / Pearson Education Inc., 2013.
2. Unix System Programming using C++, T.Chan, PHI, 1999

References:

1. Unix Network Programming, W R Stevens, PHI, 2003.
2. Unix Internals: The New Frontiers, Uresh Vahalia, Pearson Education, 1995
3. Unix for Programmers and Users, Graham Glass and King Ables, 3rd Edition, Pearson Education, 2003.

Course Code: 22HS401/22HS301/22HS351

BUSINESS ECONOMICS AND FINANCIAL ANALYSIS

(Common to ALL)

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 obtain knowledge about types of Business Structures and features.
2: To learn various principles of Managerial Economics and to make them effective business decision makers.
3: To make the students understand functional areas and potential problems in economics for efficient utilization of resources.
4: To have an overview on market structures and competition.
5: To gain knowledge on important elements.

Unit I – Introduction to Business and Economics

Business: Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance.
Economics: Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply and Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

Unit II - Demand and Supply Analysis

Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making.
Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.
Supply Analysis: Determinants of Supply, Supply Function and Law of Supply.

Unit III - Production, Cost, Market Structures & Pricing

Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale.
Cost analysis: Types of Costs, Short run and Long run Cost Functions.
Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Mono polistic Competition.
Pricing: Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis (Simple Problems)

Unit IV – Financial Accounting

Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts (Simple Problems).

Unit V – Financial Ratios Analysis

Concept of Ratio Analysis, Importance and Types of Ratios, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios –Analysis and Interpretation (simple problems).

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

CO 1	:	Understand the concepts of micro and macro economics
CO 2	:	Analyze demand and forecast demand
CO 3	:	Evaluate markets and formulate competitive strategies
CO 4	:	Prepare financial statements
CO 5	:	Evaluate the financial strengths and weaknesses of a business by using ratio analysis

Textbooks:

1 : Business Economics - Theory and Applications, D. D. Chaturvedi, S. L. Gupta, International Book House Pvt. Ltd. 2013.

2 : Financial Accounting, Dhanesh K Khatri, Tata Mc –Graw Hill, 2011.

3 : Managerial Economics, Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, 2nd Edition, Tata McGraw Hill Education Pvt. Ltd. 2012.

References:

1. Financial Accounting for Management, Paresh Shah, 2nd Edition, Oxford Press, 2015.
2. Financial Accounting, S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, 5th Edition, Vikas Publications, 2013.

Course Code: 22IT403

QUANTUM COMPUTING

(Professional Elective-III)
(Common to IT and CSE)

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

Prerequisites: A course on "Data Structures"

Course Objectives:

1: To Understand the basic Concepts of Linear Algebra related to Quantum Computing.

2: To Master the basics of physics-oriented phenomenon related to Quantum Computing.

3: To get familiar with various Quantum Architecture and Quantum Algebra.

Unit - I Introduction to Essential Linear Algebra

Some Basic Algebra, Matrix Math, Vectors and Vector Spaces, Set Theory. Complex Numbers: Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers. Database System, The Journey to Quantum Computing.

Unit - II Basic Physics for Quantum Computing

Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement.
Basic Quantum Theory: Further with Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram Quantum Entanglement and QKD, Quantum Entanglement, Interpretation, QKE.

Unit - III Quantum Architecture

Further with Qubits, Quantum Gates, More with Gates, Quantum Circuits, The D-Wave Quantum Architecture. Quantum Hardware: Qubits, How Many Qubits Are Needed? Addressing Decoherence, Topological Quantum Computing, Quantum Essentials. Introduction to Query Language

Unit - IV Quantum Algorithms

Introduction, Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Shor's Algorithm, Grover's Algorithm

Unit - V Current Asymmetric Algorithms

RSA, Diffie-Hellman, Elliptic Curve. The Impact of Quantum Computing on Cryptography: Asymmetric Cryptography, Specific Algorithms, Specific Applications.

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

CO 1	:	Understand basics of quantum computing
CO 2	:	Understand physical implementation of Qubit
CO 3	:	Understand Quantum algorithms and their implementation
CO 4	:	Understand the Impact of Quantum Algorithms and its significance
CO 5	:	Realize the importance of current Asymmetric Algorithms on Quantum computing

Textbooks:

1. Quantum Computation and Quantum Information, Nielsen M. A., Cambridge University Press
2. Quantum Computing Fundamentals, Dr. Chuck Easttom, Pearson

References:

1. Quantum Computing for Computer Scientists, Noson S. Yanofsky and Mirco A. Mannucci, 1st Edition, Cambridge University Press, 2008.
2. Principles of Quantum Computation and Information, Benenti G., Casati G. and Strini G., Vol. Basic Tools and Special Topics, World Scientific Publishing Co Pte Ltd, 2007
3. An Introduction to Quantum Computing Algorithms, Pittenger A. O., Springer-Verlag New York Inc, 2012.

Course Code: 22IT404

CLOUD SECURITY

(Professional Elective-III)
(Common to IT, CSE, CSE(CS))

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

Prerequisites: Fundamentals of cyber security

Course Objectives:

1: Understand the fundamentals of cloud computing architectures based on current standards, SLA, and cloud security models to integrate.
2: Understand the industry security standards, regulatory mandates, audit policies and compliance requirements for Cloud based infrastructures.
3: Approaches to designing cloud services that meets essential Cloud infrastructure security and management.
4: Identify the known threats, risks, vulnerabilities, and privacy issues associated with Cloud based IT services and IAM services.

Unit -I Introduction to cloud computing and cloud security

Cloud Computing Architectural Framework: Cloud Computing Evolution, Essential Characteristics of Cloud Computing, Cloud Deployment Architecture, Cloud Deployment models, Cloud Service Models, SLA.

Introduction to Cloud Security: Introduction, Cloud Security Concepts, Cloud security Standards, CSA Cloud Reference Model, NIST Cloud Reference model.

Unit - II Compliance, audit and privacy issues

Compliance and Audit: Cloud customer responsibilities, Compliance and Audit Security Recommendations. Portability and Interoperability: Changing providers reasons, Changing providers expectations, Internal Policy compliance, governance risk compliance(GRC),

Cloud Security and Privacy Issues: Introduction, Cloud Security Goals/Concepts, Cloud Security Issues, Security Requirements for Privacy, Privacy Issues in Cloud.

Unit - III Cloud infrastructure security

Identity and Access Management (IAM) in cloud environments, Virtual Private Cloud (VPC) and network security, Secure provisioning and configuration management in the cloud. The Network Level, the Host Level, the Application Level. Security Management In Cloud, Availability management for SAAS, PAAS, IAAS.

Unit - IV Threat Model and Cloud Attacks

Introduction, Threat Model- Type of attack entities, Attack surfaces with attack scenarios, A Taxonomy of Attacks, Attack Tools-Network-level attack tools, VM-level attack tools, VMM attack tools, Security Tools, VMM security tools

Unit - V Identity and Access Management

The role of IAM in cloud security, Key concepts: authentication, authorization, and auditing, oAuth and SSO, User identity and access lifecycle management, Configuring IAM in cloud platforms (e.g., AWS, Azure). **Authorization and Role-Based Access Control (RBAC)** Role-based access control principles, Creating and managing roles in cloud platforms, Least privilege principle.

Identity Federation- Understanding identity federation, Cross-domain and cross-cloud identity integration, Use cases and challenges, Virtualization, Virtualization Security Recommendations.

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

CO1 : Demonstrate the growth of Cloud computing and cloud security, architecture, and different modules of implementation

CO2 : Evaluate the different types of cloud Compliance and Audit, Security and Privacy Issues.

CO3 : Access the security implementation flow, actions and responsibilities of stake holders using IAM

CO4 : Able to analyze the various threats and Attack tools details

CO5 : Able to implement based on roles and groups policy created and choose the type of virtualization to be used

Textbooks:

1. Cloud Security and Privacy, An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, Oreilly Media, 2009.
2. Cloud Security Attacks, Techniques, Tools, and Challenges, Preeti Mishra, Emmanuel S Pilli, Jaipur R C Joshi Graphic Era, 1st Edition, CRC press, 2022.

References:

1. Securing the Cloud, Cloud Computer Security Techniques and Tactics, Vic(J.R.), Winkler, Syngress, 2011.
2. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP)

Course Code: 22CY405

BLOCKCHAIN TECHNOLOGIES

(Professional Elective-III)
(Common to CSE (CS), CSE, IT)

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: Conceptual understanding of the function of Blockchains as a method of securing distributed ledgers, how consensus on their contents is achieved.

2: To understand blockchain operations as distributed data structures and decision-making systems.

3: To evaluate "smart contract" capabilities and platforms, and examines their future directions, opportunities, risks, and challenges.

Unit - I - Introduction to Blockchain

The growth of Blockchain technology, Distributed systems, History of Blockchain and Bitcoin, Types of Blockchain, Consensus, CAP theorem and Blockchain**.**

Decentralization: Decentralization using block chain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization, Smart contracts, Decentralized Organizations, Platforms for decentralization.

Unit - II Bitcoins

Introducing Bitcoin, Digital keys and addresses, Transactions**,** The structure of a block, Mining.

Bitcoin Network and Payments: The Bitcoin network, Wallets, Bitcoin payments, Innovation in Bitcoin, Bitcoin Clients, and APIs, Bitcoin installation, Alternative Coins, Bitcoin limitations.

Unit - III Smart contracts

History, Definition, Ricardian contracts, Introduction to Ethereum, Components of the Ethereum ecosystem, Further Ethereum, Programming languages.

Unit - IV Ethereum Development Environment

Test networks, Development Tools and Frameworks, Compilers, Solidity compiler (solc), Integrated Development Environments (IDEs).

Solidity language: Layout of a Solidity, Data Types: Reference types, Value types; Literals, Enums, Function types, Global variables, Control structures.

Unit - V Hyper ledger and beyond cryptocurrency

Projects under Hyperledger, Hyperledger as a protocol, the reference architecture, Requirements and design goals of Hyperledger Fabric, Hyperledger Fabric, Membership services, Blockchain services, Consensus services, Distributed ledger.

Beyond Cryptocurrency: Applications of blockchain in cyber security, integrity of information, E-Governance and other contract enforcement mechanisms, Limitations of blockchain as a technology, and myths vs. reality of blockchain technology.

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

CO 1 : Understand the structure of a blockchain and how it is better than a simple distributed database

CO 2 : Evaluate the blockchain based structure along with its potentials and its limitations

CO 3 : Understand smartǁ contract and what are its legal implications and what it can and cannot do

CO 4 : Attain awareness of the new challenges around blockchains and smart contracts

CO 5 : Understand the differences between the different blockchain structures and their specific uses

Textbooks:

1. Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, Imran Bashir, 3rd Edition, Packt Publishing, 2020.
2. Mastering Blockchain, Imran Bashir, Second Edition, Packt Publishing, March 2018.
3. Mastering Bitcoin Programming the Open Blockchain, Andreas M. Antonopoulos, 2nd Edition, O'Reilly Media, Inc., June 2017.

References:

1. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric.
2. Hyperledger Tutorials - https://www.hyperledger.org/use/tutorials
3. Ethereum Development Resources - https://ethereum.org/en/developers

Course Code: 22DT406

NoSQL Databases

(Professional Elective – III)
(Common to CSE(DS), CSE, IT)

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: Define, compare and use the four types of NoSQL Databases (Document-oriented, key-value pairs, Column-oriented and Graph).

2: Demonstrate an understanding of the detailed architecture, define objects, load data, query data and performance-tune Column-oriented NoSQL databases.

3: Explain the detailed architecture, define objects, load data, query data and performance-tune Document-oriented NoSQL databases.

Unit - I Introduction to NoSQL

Why NoSQL? The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, A (Mostly) Standard Model, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL, Aggregate Data Models; Aggregates, Example of Relations and Aggregates, Consequences of Aggregate Orientation, Key-Value and Document Data Models, Column-Family Stores, Summarizing Aggregate-Oriented Databases. More Details on Data Models; Relationships, Graph Databases, Schemaless Databases, Materialized Views, Modeling for Data Access.

Unit - II Distribution Models

Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication. Consistency, Update Consistency, Read Consistency, Relaxing Consistency, The CAP Theorem, Relaxing Durability, Quorums. Version Stamps, Business and System Transactions, Version Stamps on Multiple Nodes.

Unit - III Map-Reduce

Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, A Two Stage Map-Reduce Example, Incremental Map-Reduce Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preference, Shopping Cart Data, When Not to Use, Relationships among Data, Multioperation Transactions, Query by Data, Operations by Sets.

Unit - IV Document Databases

What is a Document Database? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Event Logging, Content Management Systems, Blogging Platforms, Web Analytics or Real-Time Analytics, E-Commerce Applications, When Not to Use, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure.

Unit - V Graph Databases

What is a Graph Database? Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Connected Data, Routing, Dispatch, and Location-Based Services, Recommendation Engines, When Not to Use.

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

CO 1 : Acquire the knowledge of NoSQL databases
CO 2 : Articulate the distributed models
CO 3 : Construct the database using MapReduce model
CO 4 : Illustrate the document databases
CO 5 : Distinguish the graph databases with respect to social networks

Textbooks:

1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Sadalage, P. & Fowler, Wiley Publications, 2019

References:

1. NoSQL For Mere Mortals, Dan Sullivan, 1st Edition, Pearson Education India, 2015.
2. Making Sense of NoSQL: A guide for Managers and the Rest of us,Dan McCreary and Ann Kelly, 1st Edition, Manning Publication/Dreamtech Press, 2013.
3. MongoDB: The Definitive Guide- Powerful and Scalable Data Storage, Kristina Chodorow, 2nd Edition, O'Reilly Publications, 2013.

Course Code: 22AM408/22AM305

AUGMENTED REALITY AND VIRTUAL REALITY

(Professional Elective – III)
(Common to CSE(AI&ML), CSE, IT)

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: The objective of this course is to provide a foundation to the fast-growing field of AR and make the students aware of the various AR devices.

2: To give historical and modern overviews and perspectives on virtual reality. It describes the fundamentals of sensation, perception, technical and engineering aspects of virtual reality systems.

Unit - I Introduction to Augmented Reality

What Is Augmented Reality - Defining augmented reality, history of augmented reality, The Relationship Between Augmented Reality and Other Technologies-Media, Technologies, Other Ideas Related to the Spectrum Between Real and Virtual Worlds, applications of augmented reality Augmented Reality Concepts- How Does Augmented Reality Work? Concepts Related to Augmented Reality, Ingredients of an Augmented Reality Experience.

Unit -II AR Devices and Components

AR Components – Scene Generator, Tracking system, monitoring system, display, Game scene. AR Devices – Optical See- Through HMD, Virtual retinal systems, Monitor bases systems, Projection displays, Video see-through systems.

Unit - III Introduction to Virtual Reality

Defining Virtual Reality, History of VR, Human Physiology and Perception, Key Elements of Virtual Reality Experience, Virtual Reality System, Interface to the Virtual World-Input & output- Visual, Aural & Haptic Displays, Applications of Virtual Reality

Unit - IV Representing the Virtual World

Representation of the Virtual World, Visual Representation in VR, Aural Representation in VR and Haptic Representation in VR, Case Study: GHOST (General Haptics Open Software Toolkit) software development toolkit.

Unit - V Visual Perception and Rendering

Visual Perception - Perception of Depth, Perception of Motion, Perception of Color, Combining Sources of Information, Visual Rendering -Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates.

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

CO 1	:	Describe how AR systems work and list the applications of AR.
CO 2	:	Understand and analyze the hardware requirement of AR.
CO 3	:	Describe how VR systems work and list the applications of VR.
CO 4	:	Understand the design and implementation of the hardware that enables VR systems to be built.
CO 5	:	Understanding visual perception and rendering mechanisms.

Textbooks:

1. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016.
2. AR Game Developmentǁ, Allan Fowler, 1st Edition, A press Publications, 2018

References:

1. Understanding Virtual Reality: Interface, application, and design., Sherman, W. R., & Craig, A. B. 2018.
2. Augmented Reality: Principles & Practice, Schmalstieg / Hollerer, Pearson Education India; First edition, 2016.

Course Code: 22CS402

STORAGE MANAGEMENT SYSTEMS

(Professional Elective – IV)

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 information growth and challenges, storage systems and Data centers.

2 : To introduce Network-Attached Storage and intelligent storage systems.

3 : To solve the optimal route establishment problems for data delivery using relevant metrics.

4 : To serve data at the end point level and to ensure reliable data delivery mechanisms.

5 : To gain knowledge on important elements. To model secured exchange of high-level data between two applications

Unit I - Introduction to Information Storage and Data Center

Evolution of Storage Architecture, Data Center Infrastructure, Virtualization and Cloud Computing.

Data Center Environment: Application, Database Management System (DBMS), Host(compute),Connectivity, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Introduction to Flash Drives .
Data Protection: RAID Implementation Methods, RAID Array Components, RAID Techniques, RAID Levels, RAID Impact on Disk Performance, RAID Comparison.

Unit II - Intelligent Storage System

Components of an Intelligent Storage System, Storage Provisioning, Types of Intelligent Storage Systems.

Network-Attached Storage: General-purpose Servers verses NAS Devices, Benefits of NAS, File Systems and Network File Sharing, Components of NAS, NAS I/O Operation, NAS Implementations, NAS File-Sharing Protocols, Factors Affecting NAS Performance, File-Level Virtualization.

Unit III - Fibre Channel Storage Area Networks

The SAN and its Evolution, Components of FC SAN, FC Connectivity, Switched Fabric ports, Fibre Channel Architecture, Fabric Services, Switched Fabric Login Types, Zoning, FC SAN Topologies, Virtualization in SAN.

IP SAN and FCoE: iSCSI, FCIP, FCoE.

Unit IV - Object-Based and Unified Storage

Object-Based Storage Devices, Content-Addressed Storage, CAS Use Cases, Unified Storage.
Introduction to Business Continuity: Information Availability, BC Terminology, BC Planning Life Cycle, Failure Analysis, Business Impact Analysis, BC Technology Solution.

Unit V - Backup and Archive

Backup Purpose, Backup Considerations, Backup Granularity, Recovery Considerations, Backup Methods, Backup Architecture, Backup and Restore Operations, Backup Topologies, Backup in NAS Environments, Backup Targets, Data Deduplication for Backup, Backup in Virtualized Environments, Data Archive, Archiving Solution Architecture.

Managing the Storage Infrastructure: Monitoring the storage infrastructure, Storage Infrastructure Management Activities, Storage Infrastructure Management Challenges, Developing an Ideal Solution, Information Life cycle Management, Storage Tearing.

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

CO 1 : Describe the large volume data storage systems and its privacy.
CO 2 : Design the intelligent storage systems and network attached storage.
CO 3 : Illustrate the Internet Protocol based storage area networks embedding.
CO 4 : Demonstrate the business storage devices and solutions.
CO 5 : Implement Backups and maintain to protect businesses data.

Textbooks:

1. Information Storage and Management: Storing, Managing and Protecting Digital Information in Classic, Virtualized and Cloud Environment, Somasundaram Gnanasundaram, Alok Shrivastava, 2nd Edition, EMC Corporation, Wiley publication, 2012.

References:

1. Building Storage Networks, Marc Farley, 2nd Edition, Tata McGraw Hill/Osborne, 2001.
2. Storage Area Network Fundamentals, Meeta Gupta, 1st Edition, Pearson Education, 2002.
3. Building Storage Networks, Marc Farley, Tata McGraw Hill, Osborne, 2001.
4. Storage Networks: The Complete Reference, Robert Spalding, Tata McGraw Hill, Osborne, 2003.

Course Code: 22CS403

ENTEREPRISE APPLICATIONS AND ARCHITECTURAL PATTERNS

(Professional Elective – IV)

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 identify Functional Areas of an Enterprise
2: To propose Enterprise Architectural Solutions for Functional Areas
3: To apply Suitable Design Patterns in Enterprise Applications
4: To explain Enterprise Application Frameworks

Unit I - Layering and Mapping

Layering: The Evolution of Layers in Enterprise Applications, The Three Principal Layers, Choosing Where to Run Your Layers.

Mapping to Relational Databases: Architectural Patterns, The Behavioral Problem, Reading in Data, Structural Mapping Patterns, Mapping Relationships, Inheritance, Building the Mapping, Double Mapping, Using Metadata, Database Connections, Some Miscellaneous Points

Unit II - Concurrency

Concurrency Problems, Execution Contexts, Isolation and Immutability, Optimistic and Pessimistic Concurrency Control, Preventing Inconsistent Reads, Deadlocks, Transactions, ACID, Transactional Resources, Reducing Transaction Isolation for Liveness, Business and System Transactions, Patterns for Offline Concurrency Control, Application Server Concurrency.

Unit III - Distribution and logic patterns

Distribution Strategies: The Allure of Distributed Objects, Remote and Local Interfaces, Where You Have to Distribute, Working with the Distribution Boundary, Interfaces for Distribution.

Domain Logic Patterns: Transaction Script, the Revenue Recognition Problem, Domain Model, Table Module Service Layer. Data Source Architectural Patterns: Table Data Gateway, Row Data Gateway, Active Record, Data Mapper

Unit IV - Behavioral and structural patterns

Object-Relational Behavioral Patterns: Unit of Work, Identity Map, Lazy Load.

Object-Relational Structural Patterns: Identity Field, Foreign Key Mapping, Association Table Mapping, Dependent Mapping, Serialized LOB, Single Table Inheritance, Class Table Inheritance, Concrete Table Inheritance, Inheritance Mappers, Object-Relational Metadata Mapping Patterns: Metadata Mapping, Query Object, Repository.

Unit V - Web and offline patterns

Web Presentation Patterns: Model View Controller, Page Controller, Front Controller, Template View, Transform View, Application Controller. Distribution Patterns: Remote Facade, Data Transfer Object.

Offline Concurrency Patterns: Optimistic Offline Lock, Pessimistic Offline Lock, Coarse-Grained Lock, Implicit Lock.

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

CO 1 : Understand the Layering of Enterprise applications and their mapping to Relational Databases.
CO 2 : Illustrate the Concurrency Control mechanisms in Enterprise applications
CO 3 : Demonstrate the Distribution Strategies and Domain Logic Patterns.
CO 4 : Understand and implement the Object-Relational Behavioral Patterns.
CO 5 : Implement Web Presentation Patterns and Distribution Patterns.

Textbooks:

1. Patterns of Enterprise Application Architecture, Martin Fowler,1st Edition, Addison-Wesley Education, 2002.

References:

1. The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise, Gregor Hohpe, 1st Edition, Shroff/O'Reilly, 2020.
2. Simple SysML for Beginners: Using Sparx Enterprise Architect, David Hetherington, 1st Edition Asatte Press, 2019.
3. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Gregor Hohpe, Bobby Woolf, 1st Edition, Pearson Addison-Wesley Professional, 2003.

Course Code: 22IT405

INTERNET OF EVERYTHING (IoE)

(Professional Elective-IV)
(Common to IT, CSE)

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

Prerequisites: Internet of Things (IoT)

Course Objectives:

1 : Understanding of Industrial IOT Infrastructure and Issues.
2 : Monitoring of End Devices.
3 : Understanding of Smart City Environment.
4 : Familiarity of Internet of Medical Things.

Unit - I Industrial IOT
Definition, IoT v IIoT, Next Generation Sensors, Sensor's calibration and validate sensor measurements, placement of IoT devices, sensors, low-cost communication system design, Top application areas include manufacturing, oil & gas, Embedded systems in the Automotive and Transportation market segment.

Unit - II Monitoring of end devices
Enablement of data-driven digital equipment model to monitor assets and systems, Introduction to device localization and tracking; different types of localization techniques, Radio-Frequency Identification (RFID) and fingerprinting, Device diversity/heterogeneity issue in IIoT networks

Unit - III Implementation of industrial IoT using AIML
Data flow, big data and how to prepare data for machine learning algorithms, Machine Learning algorithms, supervised learning & Un-supervised learning algorithms, Basics of neural network, activation functions, back-propagation.

Unit - IV Smart City
IoT in smart city& their distinctive advantages like smart environment, smart streetlight, smart water management, Smart Road & Traffic, Smart Parking & waste management. The Fourth Revolution, LEAN Production Systems, Smart and Connected Business Perspective, Smart Factories.

Unit - V Internet of Medical Things
Introduction to IoT applications in smart healthcare& their distinctive advantages - Patient Health Monitoring System (PHMS), Mobile Health Things (m-health).

Emerging Technologies for Health and Medicine: Virtual Reality, Augmented Reality, Artificial Intelligence, Robotics, Industry 4.0.

Course Outcomes: The student will be able to

CO 1 : Gain knowledge of Industrial sensors and Embedded Systems
CO 2 : Usage of IOT data for ML
CO 3 : Able to connect and communicate with different IOT devices
CO 4 : Collecting and displaying Smart City data in proper format
CO 5 : Monitoring and control of patient health using Medical IOT

Textbooks:

1. Industry 4.0: The Industrial Internet of Things, Alasdair Gilchrist, Apress, 2016.
2. Designing, Developing, and Facilitating Smart Cities Urban Design to IoT Solutions, Vangelis Angelakis, Springer, 2019
3. Emerging Technologies for Health and Medicine: Virtual Reality, Augmented Reality, Artificial Intelligence, Internet of Things, Robotics, Industry 4.0, Dac-Nhuong Le, Wiley, 2019.

References:

1. Introduction to Industrial Internet of Things and Industry 4.0, Sudip Misra, Chandana Roy, Anadarup Mukherjee, CRC Press,2021
2. Introduction to IoT, S. Misra, A. Mukherjee, and A. Roy, Cambridge University Press, 2017

Course Code: 22CS404

DATA WAREHOUSING AND DATA MINING

(Professional Elective – IV)

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 demonstrate the value of Data mining in solving real-world problems.

2: Demonstrate Understanding of foundational concepts underlying Data mining.

3: Demonstrate Understanding of algorithms commonly used to perform various Data mining tasks.

4: To Learn and understand the cluster analysis and clustering methods used in data mining.

5: To Learn and understand the Time series, text and web mining applications and data mining trends, challenges, and its applications.

Unit I – Introduction to Data Mining and Data Warehouse

Fundamentals of Data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining Task Primitives, Data Warehouse, Integration of a Data Mining System with a Database or a Data Warehouse System, Multidimensional Data Model, A three tier Data Warehouse Architecture, OLAP Technology for Data Mining.

Unit II – Association Rule Mining

Data Characterization, Data Discrimination, Attribute-Oriented Induction.
Association Rule Mining: Basic Concepts, Efficient and Scalable Frequent Itemset Mining methods. Mining various kinds of Association rules, from Association Analysis to Correlation Analysis, Constraint-based Association Mining.

Unit III - Classification and Prediction

Classification: Classification by Decision Tree Induction, Bayesian Classification, Rule based classification, Eager learners, Lazy Learners, Soft computing approaches -rough set, Fuzzy logic.
Accuracy and Error measures: Evaluating the accuracy of a classifier or a predictor, Ensemble Methods.

Unit IV – Cluster Analysis

Introduction to Cluster Analysis: Types of Data in Cluster Analysis, A Categorization of **Major Clustering Methods**: Hierarchical Method-BIRCH, Density-Based Methods- DBSCAN and DENCLUE, Grid-Based Methods- STING, Model-Based Clustering Methods-Expectation Maximization
Clustering High-Dimensional Data: PROCLUS, Outlier Analysis.

Unit V – Time Series, Text and Web Mining

Mining Time-series data: mining sequence patterns in Transactional Databases, Text Mining, Mining the World Wide Web, VIPS and HITS algorithms.
Applications and Trends in Data Mining: Data Mining Applications, Major issues and challenges in Data Mining, Social Impacts of Data Mining.

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

CO 1	:	Understand different data mining tasks and apply the algorithms most appropriate for addressing them.
CO 2	:	Discover and Analyze interesting patterns from different kinds of databases.
CO 3	:	Apply the techniques of classification and prediction to build and use supervised learning from datasets.
CO 4	:	Apply the techniques of clustering to implement unsupervised learning systems.
CO 5	:	Understand nature of time-series, web and text data to develop methodologies and application for such data analysis and mining.

Textbooks:

1. Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber and Jian Pei, 3rd Edition, Morgan Kaufmann Publishers/Elsevier, 2011.

References:

1. Data Mining Techniques, Arun K Pujari, 2nd Edition, University Press, 2013.
2. Introduction to Data Mining, Pang Ning Tan, Michael Steinbach and Vipin Kumar, Pearson Education, 2007.
3. Data Warehousing in the Real World, Sam Anahory and Dennis Murray, Pearson Education Asia, 2011.

Course Code: 22CS405

DEEP LEARNING AND APPLICATIONS

(Professional Elective-IV)

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 develop an understanding of the basic concepts of Artificial Intelligence.
2 : To gain the rationale behind various discriminative and generative models of ANN
3 : To impart knowledge on combining the capabilities of different models
4 : To appreciate the encoder-decoder architectures that solve linguistics related problems.

Unit I –Multilayer Perceptron and Back Propagation

Introduction: Perceptron, Learning Algorithm, Feed forward Neural networks, Gradient descent and the back propagation algorithm, Unit saturation, the vanishing gradient problem, and ways to mitigate it. Various Activation Functions, ReLU Heuristics for avoiding bad local minima, Heuristics for faster training, Momentum, RMSProp, ADAMS, Nestors accelerated gradient descent, Bias-Variance trade off, Regularization, Dropout, weight initialization techniques – Random, He, Xavior.

Unit II – Deep Learning Models

Convolutional Neural Networks: Architectures, convolution/pooling/stride layers, Feature maps, Fully Connected Layer, Training. Transfer Learning, ALEXNet, VGG, ResNet.
Auto Encoders: Architecture, Training, Drawbacks, Denoising Autoencoders
Restricted Boltzmann Machine: Architecture, Training, Contrastive Divergence
Variational Auto Encoders: Architecture, Training, ELBO, KL- Divergence, Reparameterization.
Adversarial Generative Networks: Architecture, Cross-Entropy and MSE, MINMAX algorithm, Challenges in training

Unit III - Applications of Deep Learning to Computer Vision

Image segmentation, object detection, Image Denoising, Face Recognition, Image Reconstruction using VAE, Image generation with Generative adversarial networks, Object Tracking, YOLO4 for computer vision tasks.

Unit IV - Vector Space Model of Semantics, Sequence Models

Vector Representation of Words: one-hot-encoding, BoW, TF-IDF and information retrieval, Co-Occurrence Matrix, SVD, Word similarity.
Distributed Representations: Word2Vec, Continuous Skip-Gram Model, Continuous Bag-of-Words model (CBOW), GloVe, Evaluations and Applications in word similarity.
Sequence Models: Sequence Learning, Drawbacks of CNN, RNN Architecture, Template architectures, BPTT algorithm, Limitations of RNN, LSTM -architecture, Activation functions, gates, functions, training, LLMs- Encoder and Decoder based models, Transformer and BERT.

Unit V – Applications of Deep Learning to NLP

Analogy reasoning: Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks, Dialogue Generation with LSTMs, Image Captioning R-CNN, Attention models. Exposure to RASA and ChatGPT.

Course Outcomes: After completion of course, students will be able to:

CO 1 : Implement Backpropagation algorithm and solve non-liner classification problem.

CO 2 : Solve image classification problem using transfer learning.

CO 3 : Derive a functionality from a generative model to create fake images, restore the images.

CO 4 : Handle the complexities of Natural Language processing and gain proficiency in designing language models.

CO 5 : Solve real world problems using CNN, RNN, and LSTM based RASA framework.

Textbooks:

1. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016
2. The Elements of Statistical Learning, T. Hastie, R. Tibshirani, and J. Friedman, Springer, Second Edition, 2009.

References:

1. Pattern Recognition and Machine Learning, Bishop, C, M., Springer, 2006.
2. Artificial Neural Networks, Yegnanarayana, B., PHI Learning Pvt. Ltd, 2009.
3. Matrix Computations Golub, G.H., and Van Loan,C.F, JHU Press, 2013.
4. Neural Networks: A Classroom Approach, Satish Kumar, Tata McGraw-Hill, 2004.

Course Code: 22CS431

BIG DATA ANALYTICS AND PLATFORMS LAB

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 gain familiarity with distributed file systems and scalable computing paradigms.
2 : To derive actionable knowledge from big data.
3 : To handle stream data and implement predictive analytics.
4 : To implement scalable and resilient event-driven architectures.

Task 1:

1. Configure and Install Hadoop in Pseudo-Distributed Mode.
2. Using the command line interface, execute a series of operations on the Hadoop Distributed File System (HDFS):
    1. Copy Files to HDFS
    2. Copy Files from HDFS
    3. List the directories and files in HDFS
    4. Display the contents of file in HDFS
    5. Create a directory in HDFS
    6. Remove a directory/file in HDFS
    7. Count the number of files in HDFS
    8. Merge files in HDFS
    9. Concatenate two files and store them in HDFS
    10. Find the checksum of the file in HDFS

Task 2:

1. Experiment importing and exporting data into and from HDFS using Apache Sqoop
2. Demonstrate the utilization of Apache Flume for collecting log files generated by any application server and subsequently loading of the data into HDFS

Task 3:

Develop a Map Reduce program to calculate the frequency of a word in a file.

Task 4:

Develop a MapReduce program to analyze Uber dataset to find the days on which each basement has more trips using the following dataset.

The Uber dataset consists of four columns

Dispatching_base_number	Date	Active_vechiles	Trips

Task 5:

1. Conduct the following operations within Hive:
    1. Create a Datastore
    2. Listing Datastores
    3. Switching Datastore
    4. Viewing the current Datastore
    5. Dropping a Datastore
    2. Use the following schema structure and construct tables using HiveQL.

Sales Order Table: sales_order

order_id INT,
customer_id INT,

product_id INT,
order_date TIMESTAMP,
order_amount DOUBLE,
quantity INT,
discount DOUBLE,
tax DOUBLE,
total_amount DOUBLE
PRIMARY KEY (order_id)

Product Table: product
product_id INT,
product_name STRING,
category STRING,
price DOUBLE,
manufacturer STRING,
date_added TIMESTAMP,
PRIMARY KEY (product_id)

Customer Table: customer
customer_id INT,
customer_name STRING,
email STRING,
phone STRING,
address STRUCT<street: STRING, city: STRING, state: STRING, zip: INT>,date_joined TIMESTAMP,PRIMARY KEY (customer_id)PARTITIONED BY (country STRING, state STRING);

a) Find Total Sales Amount
b) Find Top N product by Sales
c) Find Customer-wise Sales
d) Find Monthly Sales Trends
e) Find Product-wise Quantity Sold
f) Find the Highest-Priced Product in Each Category

Task 6:

1. Demonstrate the usage of Hive User Defined Function (UDF) to manipulate data in a Hive table.

Task 7:

Use the following schema structure and construct HBase tables:

Table Name: employee_details
Column Families: PersonalInfo, ProfessionalInfo
PersonalInfo: name, age, gender
ProfessionalInfo: designation, salary, department

1. Create employee_details HBase table
2. Insert ten records into the employee_detailstable
3. Display details of an employee with a specific name
4. Update the Salary of an employee with a specific name
5. Delete the record of an employee with a specific name

Task 8:

1. Demonstrate Hive and HBase for seamless data transfer between two storage systems with an example also verify in HBase.

Task 9:

1. Consider an Employee dataset that includes fields such as Employee ID, Name, Department, Salary, and Year of Joining. Apply the principles of the Resilient Distributed Dataset (RDD) to perform operations on this dataset.

   a) Load the Dataset into an RDD

b) Count the number of records in the RDD
c) Filter out employees who joined after 2014
d) Map the RDD to a new RDD with only employee names
e) Calculate the average salary of employees for each department

Task 10:

1. Write a PySpark program to read an employee_data.csv file, store it in DataFrame, and perform a series of operations on the DataFrame:
    1. Calculate the average salary for each department.
    2. Create a new column 'Salary Increase' that shows the percentage increase in salary for each employee compared to the average salary in their department
    3. Determine the number of years each employee has been with the company and add this information as a new column 'YearswithCompany'
    4. Create a new column 'Salary Category' that categorizes employees into different salary ranges (Low, Medium, High)
2. Write a PySpark program to ingest data from a database, store it in a Spark DataFrame, and subsequently execute the following operations:

    1. Get the details about the DataFrame.
    2. Check for Missing Values
    3. Replace Missing Values with Mean/Median/Most frequent items of the column.
    4. Save the data back to the Database.

Task 11:

1. Develop an experiment to showcase real-time data processing capabilities using the Apache Spark Streaming API utilizing any input source such as socket, file, or Kafka.

Task 12:

1. Utilize Spark ML API and build a Linear Regression Model to predict the house price for a Boston Housing Prices dataset.
2. Write a program to demonstrate the working of the decision tree-based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Task 13:

1. Understand the GraphX Modeling of Page Rank algorithm using PySpark

Task 14:

1. Write a PySpark program to perform analytics on the employee_details table stored in HBase.

Task 15:

1. Set up Kafka locally, create topics and channels, and implement a simple event-handling system with subscribers.

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

CO 1 : Distribute data across HDFS and solve Big Data problem using map-reduce framework offered by Hadoop

CO 2 : Populate the Hive data warehouse and run online analytics

CO 3 : Perform real-time data and graph analytics on stream data using Apache Spark and machine learning algorithms

CO 4 : Develop PySpark programs to connect with various storage systems

CO 5 : Setup, Configure, and Manage Apache Kafka in a local machine and implement event handling system

References:

1. HADOOP : The Definitive Guide, Tom White, 4th Edition,O'Reilly 2015.
2. Spark: The Definitive Guide, Bill Chambers and Matei Zaharia, 1st Edition, O'Reilly 2018.

Course Code: 22CS432

VISUAL PROGRAMMING LAB

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 provide hands-on experience on .NET Framework.
2: To appreciate the asynchronous event handling feature in .Net
3: To offer end-to-end program model for web application development

List of Experiments:

Task 1:

1. Demonstrate a Console Application program to validate Email-ID and username with 3 to 16 characters length and also can have all special characters using Regular Expressions.

2. String Manipulation with the String Builder and String Classes: Demonstrate some basic string manipulation using methods of both StringBuilder and String classes**.**

Task 2:

1. Working with callbacks and delegates in C#: Demonstrate the use of delegates, callbacks, and synchronous and asynchronous method invocation.

Task 3:

1. Working with Interface Inheritance: Demonstrate the interface inheritance using explicit interface Implementation.

Task 4:

Working with Inheritance:

Employee Management System

Tokyo Company wants to register the new employee to their data.

Write an application which generates EmployeeID, UserID, EmailID, Salary

Employee ID: It is the sequence of the employee in the organization.

1. Employee Sequence should be in between 1-9999
2. It should be always in 5 digits starting with 10000, if the employee sequence is 1 then Employee ID should be 10001, if sequence is 101 then Employee ID should be 10101

User ID: It is similar to EmployeeID, but first 2 letters in the first name is prefixed in UPPERCASE

Example: If firstname of the employee is Rajesh and Employee Id is 10101 then UserId is RA10101

Email ID: UserID@Tokyo.com

Salary: of the employee

1. NetCompensation of the Employee is=No. Of Years of experience*1(1 represented for 1 LAKH)
2. HRA is 10% of the salary
3. Salary=NetCompensation+HRA

Class Requirements:

Person class:

Member Name	Type
First Name	String
LastName	String
Gender	String
DOB	DateTime

Employee class:

Member Name	Type
Department	String
Location	String
EmpSequence	int
YearsOfExperience	int

This class should have **parameterised constructor** with

First name, last name, department, location, empseq, years Of Experience

Methods to implement:
Public string GenerateEmailID()
Public double CalculateSalary()

Registration class:
Members
Static in EmployeeID
Static string UserID
This class should have **parameterised constructor** with

Firstname, lastname, empseq

Methods to implement:
Public void GenerateEmployeeID()
Public void GenerateUserID()

Inheritance requirement:
Base class: Person
Child class 1: Employee class inherits Person.
Child class 2: Registration class inherits Employee

Task 5:

1. Using Reflection in C#: Demonstrate how to gather information on various types included in any assembly by using the System.Reflection namespace and some main.NET base classes.

Task 6:
Working with Assemblies:

1. Demonstrate a console application by creating a Private Assembly and use it in different applications.
2. Demonstrate how to Create a Public assembly and store it in GAC and use it in all applications.

(Use the following case study for Tasks 7 to 10)

ToDo Planner

XYZ Inc. is a leading office automation service provider. They want to develop an application using which the users can plan and manage their ToDos. As a developer you need to develop a console based application to manage the list of tasks. You need to define the implementation for scheduling a task, postpone, prepone, set priority, task completion etc. You are provided with the class and respective members description and you are supposed to implement them accordingly.

Namespace: Planning App

Use List Generic Class to Implement this Application

The class and it's members descriptions provided below and you need to define them accordingly.

1. Define a class by the name, Task and define the following private fields inside the Task class.

2. TaskID as int
3. TaskDescription as string
4. TaskStartDate as Date Time
5. TaskEndDate as Date Time
6. TaskPriority as string
7. CompletionStatus as string
8. In the Task class define a parameterized constructor to initialize the private fields to assign the parameter values accordingly. Within the parameterized constructor ensure that a unique integer value is auto-generated and assigned to TaskID private field.
9. Define a class by the name, To Do Planner with the following specifications.
10. Define a generic list collection of type Task.
11. Define the below methods accordingly.

public bool AddTask(Task obj)

This method is to add a task to the tasks generic collection of type Task. If the parameter obj is null, the method should return false, else the method should add the task details to tasks generic collection and return true.
TaskPriority can be either high or low initially completion status can be pending
public bool DeleteTask(int intTaskID)

This method should delete a task based on the task ID provided. If the deletion successful return true else return false, If no task exists with the given task ID, return false.

public bool Update TaskStatus(int intTaskID, string strStatus)

This method should update a particular task's status. If the task found and able to update the CompletionStatus, return true. If the task is not found or unable to update the CompetionStatus return false. Updated CompletionStatus can be either "Inprogress" or "Completed"

public Task GetTasks()

This method should return the list of tasks as an Array.
Use List Generic Concept to implementthisprogram
List<Task> task=new List<Task>

Task 7:

1. Sending Mail with SmtpMail : Use a simple web form to demonstrate how to use the SmtpMail class in the .Net Framework.

Task 8:

1. Using the System.NetWebClient to Retrieve or Upload Data: Demonstrate how to create windows form that can use HTTP to download and save a resource from a specified URI, upload a resource to a specified URI, or read and write data through a stream object.

Task 9:

1. Working with ASP.NET Web Pages:

Create the ASP.Net Web Application that accepts Name, Password, age, email id, and userid. All the information entry is mandatory. Password should be reconfirmed; age should be within 21-30. Email id should be valid. User id should have atleast a capital letter and digit as well as length should be between 7 and 20 characters. Use all Validation Controls.

Task 10:

1. Working with LINQ:

Create a table with given fields

Field Name	Type
EmpNo	Number
EmpName	Varchar
EmpSal	Number
EmpJob	Varchar

EmpDeptNo	Number

For the given table design a web page to display the employee information from table to grid control. Use LINQ to ADO.NET

Student Management System Application Development (For Tasks 11 to 14)
Task 11:

1. Create Website Application for Student Management System with a master page which is linked to other web pages in the application.

Task 12:

1. Create 5 content pages and design it accordingly and use different Navigation controls to navigate between content pages.

Task 13:

1. Use ADO.NET for storing and manipulating the data. Develop the necessary forms for the better user Interface.

Task 14:

1. Convert the above application to a web application using ASP.NET and SQL Server. Use IIS to deploy the web application developed in ASP.NET.

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

CO 1	:	Create Private and shared Assemblies
CO 2	:	Develop Asynchronous Applications
CO 3	:	Deploy Web Services
CO 4	:	Understand the Language Integrated Query (LINQ) Library
CO 5	:	Build Database applications using ADO.NET

References:

1. Professional C# 5.0 and .NET 4.5.1, Christian Nagel, Jay Glynn and Morgan Skinner, John Wiley & Sons Inc., 2014.
2. Beginning ASP.net 4.5.1in C# and VB, Imar Spaanjaars, Wrox Publication, 2014.
3. Microsoft Visual C# Step by Step, John Sharp, O‘Reilly Media, Inc., 2013.
4. A Tester’s Guide to .NET Programming, Randal Root and Mary Romero Sweeney, Apress, 2006.

Course Code: 22HS451/22HS404

ORGANIZATIONAL BEHAVIOUR

(Common to CSE, CSE(DS), CSE(CS))

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 nature, scope of Organizational Behavior along with basic concepts and its applications in contemporary organizations.

2 : To deeply understand the role of personality and attitude, theories of motivation, emotional intelligence.

3 : To appreciate the role and means of effective communication, decision making, strategies to reduce stress and conflict in organizations.

4 : To learn about the role of power and politics, Group dynamics, Teams in the modern workplace.

5 : To develop framework for high performance through leadership, job design, performance management and Quality of work life in organizations.

Unit - I Organisational Behaviour

Definition, need and importance of organizational behaviour – Nature and scope – Frame work – Organizational behaviour models.

Unit - II Individual Behaviour

Personality – types – Factors influencing personality – Theories – Learning – Types of learners – The learning process – Learning Theories – Organizational behaviour modification. Misbehaviour – Types – Management Intervention - Emotions – Emotional Labour – Emotional Intelligence – Theories. Attitudes – Characteristics – Components – Formation – Measurement – Values, Perceptions – Importance – Factors influencing perception – Interpersonal perception – Impression Management. Motivation – importance – Types – Effects on work behaviour.

Unit - III Group Behaviour

Organization structure – Formation – Groups in organizations – Influence – Group dynamics - Emergence of informal leaders and working norms – Group decision making techniques – Team building – Interpersonal relations – Communication – Control.

Unit - IV Leadership and Power

Meaning – Importance – Leadership styles – Theories of leadership – Leaders Vs Managers – Sources of power – Power centers – Power and Politics.

Unit - V Dynamics of Organizational Behaviour

Organizational culture and climate – Factors affecting organizational climate – Importance. Job satisfaction – Determinants – Measurements – Influence on behaviour. Organizational change – Importance – Stability Vs Change – Proactive Vs Reaction change - the change process - Resistance to change – Managing change. Stress – Work Stressors – Prevention and Management of stress – Balancing work and Life. Organizational development – Characteristics – objectives – Organizational effectiveness.

Course Outcomes: After learning this course students be able to

CO 1 : Understand the nature of Organizational Behavior and its applications in an organization

CO 2 : Motivate themselves and others with a positive approach

CO 3 : Work as a team member and build a good team

CO 4 : Be good leaders

CO 5 : Work effectively in an organization

Textbooks:

1. Organizational Behavior, Luthans, Fred, 10th Edition, McGraw-Hill, 2009.
2. Organizational Behavior, Robbins, P. Stephen, 18th Edition, Pearson, 2018.

References:

1. Organizational Behavior, Mc Shane, 9th Edition, Mc Graw Hill 2022
2. Organizational Behavior, Nelson, 3rd Edition, Thomson, 2008.
3. Organizational Behavior- Human Behavior at Work, Newstrom W. John & Davis Keith, 12th Edition, TMH, New Delhi, 2009.
4. Management and Organizational Behavior: An Integrated perspective, Pierce and Gardner, Thomson, 2009.
5. Behavioural Process at Work, Pareek Udai, Oxford & IBH, New Delhi, 2009.
6. Organizational Behaviour , Schermerhorn, 9th Edition, Wiley, 2008.
7. Organizational Behaviour, Hitt, Wiley, 2008.

Course Code: 22CS451

NATURAL LANGUAGE PROCESSING MODELS

(Professional Elective –V)

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 show how language related algorithms and techniques can be applied to important real-world problems (Spell Checking, Text Document Search, Part-of- Speech Tagging.

2: To emphasize scientific evaluation and language generation mechanisms.

3: To learn various applications of Natural Language Processing tools.

Unit I – Overview

Origins and challenges of NLP-Language and Grammar-Processing Indian Languages NLP Applications, Information Retrieval. Language Modeling: Various Grammar- based Language Models, Statistical Language Model. Tools for NLP (NLTK).

Unit II - Word Level Analysis

Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and Correction-Words and Word classes, Part-of Speech Tagging.
Syntactic Analysis: Context-free Grammar, Constituency, Parsing, Probabilistic Parsing.
Word level analysis methods in NLTK tool.

Unit III - Semantic Analysis

Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation, Discourse Processing: cohesion, Reference Resolution, Discourse Coherence and Structure.
Semantic analysis methods in NLTK tool.

Unit IV - Natural Language Generation

Architecture of NLG Systems, Generation Tasks and Representations Application of NLG.
Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Translation involving Indian Languages.

Unit V - Information Retrieval

Design features of Information Retrieval Systems, Classical, Non-classical, Alternative Models of Information Retrieval, Evaluation of the IR System. Lexical Resources: World Net, Frame Net Stemmers, POS Tagger, Research Corpora.
Information retrieval methods in NLTK.

Course Outcomes: After completion of course, students will be able to:

CO 1: Extract information from text automatically using concepts and methods from natural language processing (NLP) including stemming, n-grams, POS tagging, and parsing

CO 2: Develop speech-based applications that use speech analysis (phonetics, speech recognition, and synthesis).

CO 3: Analyze the syntax, semantics, and pragmatics of a statement written in a natural language.

CO 4: Develop a conversational agent that uses natural language understanding and generation.

CO 5: Evaluate the performance of NLP tools and systems.

Textbooks:

1. Natural Language Processing and Information Retrieval, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2023.
2. Natural Language Processing with Python: Analyzing text with Natural Language Toolkit, Steven Biord, Ewan Klein, and Edward Loper, ORielly Publications, January 2011.

References:

1. Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Daniel Jurafsky and James H Martin, 2nd Edition, Prentice Hall, 2013.
2. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python, Hobson Lane, Hannes Hapke, Cole Howard, Manning publications 2019.

Course Code: 22CS452

PENETRATION TESTING AND INCIDENT RESPONSE

(Professional Elective –V)

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 describe Cybersecurity skills.
2 : Describe penetration testing tools and the benefits to an organization.
3 : Describe digital forensics and digital evidence.
4 : To test and exploit systems using various tools.
5 : To understand the impact of hacking in real time machines

Unit I - Introduction

Penetration Testing phases/Testing Process, types and Techniques, Blue/Red Teaming, Strategies of Testing, Non-Disclosure Agreement Checklist, Phases of hacking, Open source/proprietary Pentest Methodologies

Unit II - Information Gathering and Scanning

Information gathering methodology- Foot printing, Competitive Intelligence-DNS Enumerations-Social Engineering attacks, Port Scanning-Network Scanning-Vulnerability Scanning- NMAP scanning tool- OS Fingerprinting- Enumeration.

Unit III - Preparing for the Inevitable Incident

What is Incident Response, Concept of the Attack Life cycle, Goals of Incident Response, Incident Response Process, Initial Response, Investigation, Remediation,
Pre-Incident Preparation: Identifying Risk, Policies That Promote a Successful IR, working with Outsourced IT. Preparing the IR Team: Defining the Mission, Communication Procedures, Deliverables, Resources for the IR Team

Unit IV - Incident Detection and Characterization

Collecting Initial Facts, Maintenance of Case Notes, Setting Expectations with Management, Initial Development of Leads, Turning Leads into Indicators, The Life cycle of Indicator Generation, Resolving Internal Leads, Resolving External Leads

Unit V - Live Data Collection

When to Perform a Live Response, Selecting a Live Response Tool, Live Data Collection on Unix-Based Systems, Forensic Duplication: Forensic Image Formats, Partition Image, Logical Image, Image Integrity. Traditional Duplication Network Evidence: The Case for Network Monitoring, Types of Network Monitoring, Event-Based Alert Monitoring, statistical Modelling

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

CO 1: gain Cybersecurity skills.
CO 2: understand the methodologies and techniques used for penetrating
CO 3: acquire skills to work in the Cybersecurity field as a Cybersecurity Analyst
CO 4: learn key steps in the forensic process and important data to collect
CO 5: identify security vulnerabilities and weaknesses in the target applications

Textbooks :

1. Cybersecurity Incident Response: How to Contain, Eradicate, and Recover from Incidents, Eric C. Thompson, Apress, USA, 2018
2. Incident Response & Computer Forensics, Jason T. Luttgens Matthew Pepe, Third Edition McGraw-Hill Education, 2014.

References:

1. Linux Revealed: Mastering the Penetration Testing, Raphael Hertzog, Jim O'Gorman, Kali Offsec Press, 2017.

Course Code: 22CS453

ROBOTIC PROCESS AUTOMATION

(Professional Elective - V)
(Common to CSE, CSE(AI&ML), IT)

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 the functional elements of Robotic Process Automation
2: To impart knowledge on the concepts of RPA
3: To understand the development the Bot
4: To educate on various RPA vendors
5: To understand the data preparation and process mining

Unit I - Introduction to Robotic Process Automation

RPA Foundations: Introduction to RPA, RPA Skills-web technologies, Agile, databases etc.
Process Methodologies: Lean, Six Sigma Roles and Levels, Applying Lean and Six Sigma to RPA.

Unit II – Planning

The Preliminaries, ROI for RPA, RPA Use Cases, Security, Monitoring, and Deployment, The Design, Next-Generation Technologies, RPA Solution Architect, RPA Supervisor, Change Management

Unit III -Bot Development

Preliminaries, Installation of UiPath, Flowcharts and Sequences, Log, Message, Variables, Loops and Conditionals, Debug, Common UiPath Functions, The UiPath Orchestrator, Bot Development.

Unit IV -Data Preparation

Types of Data, Big Data, The Issues with Big Data, The Data Process, Types of Algorithms.
Process Mining - Old Way Vs. Process Mining, How Process Mining Works, The Future of Process Mining.

Unit V - RPA Vendors

UiPath, IQ Bot, Bot Store, Blue Prism, EdgeVerve, PEGA.
Deployment and Monitoring- Testing, Going into Production, Monitoring, Security, Scaling.

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

CO 1	:	Understand the concepts of RPA and RPA skills lean six sigma.
CO 2	:	Design of RPA use cases and security, change management.
CO 3	:	Implement the concepts of BOT
CO 4	:	Understand the data preparation algorithms and process mining.
CO 5	:	Develop and demonstrate an application of RPA with different vendors.

Textbooks:

1. The Robotic Process Automation Handbook – A Guide to implementing RPA systems, Tom Taulli, Apress, Springer Science,2020.
2. Learning Robotic Process Automation, Alok Mani Tripathi, Packet publishing Ltd,2018.

References:

1. Robotic Process Automation for dummies, Steve Kaelbel, John Wiley and sons, 2018.
2. Designing BOTS, Amir shevat, O’Reilly, 2017.

Course Code: 22IT451

DESIGN PATTERNS

(Professional Elective-V)
(Common to IT, CSE, CSE(DS), CSE(AI&ML), CSE(CS))

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

Prerequisites:

1. Object Oriented Programming through Java
2. Software Engineering

Course Objectives:

1: To make the students understand the basic concepts of Design patterns.
2: To understand the various Design patterns.
3: To understand the importance of design patterns for development of a reusable product.

Unit I – Introduction

What Is a Design Pattern? Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern.

Unit II – A Case Study

Designing a Document Editor: Design Problems, Document Structure, Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary What to Expect from Design Patterns.

Unit III – Creational Patterns

Abstract Factory, Builder, Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

Unit IV – Structural Patterns

Adapter, Bridge and Composite, Decorator, façade, Flyweight, Proxy.

Unit V – Behavioral Patterns

Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns.

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

CO 1	:	Appreciate the basic concepts of design patterns and able to know how to select and use the design patterns
CO 2	:	Identify the design pattern in the existing code and use of creational patterns
CO 3	:	Apply and use the structural patterns
CO 4	:	Identify and use the behavioral patterns
CO 5	:	Find and catalog patterns in the object-oriented software

Textbooks:

1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley, 1995.
2. Java™ Design Patterns: A Tutorial, James W. Cooper, Addison Wesley, 2000.

References:

1. Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, Mark Grand, Volume 2, Wiley DreamTech.
2. Patterns in Java, Mark Grand, Volume 2, Wiley DreamTech, 2008.
3. Java Enterprise Design Patterns, Mark Grand, Wiley DreamTech, 2006.

Course Code: 22IT452

GENERATIVE AI

(Professional Elective-V)
(Common to IT, CSE, CSE(DS), CSE(AI&ML), CSE(CS))

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

Prerequisites: Machine learning

Course Objectives:

1 : To understand how Large Language Models (LLMs) work
2 : How to integrate generative AI into your personal and professional work
3 : Societal impacts of generative AI

Unit - I Introduction to Generative AI

Artificial Intelligence is Nothing New, Generative AI- A New Approach to AI, Human Equivalents of Predictive and Generative AI, Generative Artificial Intelligence Applications, Generative AI Models

Unit - II Introduction to LLMs

Introduction to Modern Natural Language Processing, Birth of Large Language Models, Explosion of Large Language Models, Applications of Large Language Models, Limitations and Risks of Large Language Models

Unit - III Data Privacy and Safety

Training of LLMS: Emergent Properties of LLMs, Considerations, Strategies for Improving Generations from a Safety Perspective, Data Policies and Regulations

Unit - IV Content Creation and Machine Augmented Work

Rise of Synthetic Media Generative AI in creative workflows, Intellectual Property(IP) in the LLM era, Professional Applications, Personal Usage, Detection of Machine generation, economic consequences

Unit - V Risks, Practices and Ethics of Generative AI

Prospective Developments of LLMs, Social and Technical risks of LLMs, Best practices of Responsible Use, Ethics-Informed AI regulations, Towards AI Governance Framework,

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

CO 1 : Define Generative AI and describe various Generative AI applications.
CO 2 : Explore the applications, limitations, and risks of LLMs
CO 3 : Understand how the LLMs are trained and emergent properties of LLMs
CO 4 : Leverage LLMs in professional and personal settings
CO 5 : Explore and formalize the best practices for responsible use of generative AI models

Textbooks:

1. Generative Ai - The Future Of Everything, Sharad Gandhi, Christian Ehl , 2023
2. Introduction to Generative AI :An ethical, societal, and legal overview, Numa Dhamani and Maggie Engler, , Manning Publications,2023.

Reference Books

1. Generative AI Models, Jovan Pehcevski, Arcler Press, 2024
2. Artificial Intelligence & Generative AI for Beginners: The Complete Guide, David M.Patel , GD Publishing, 2023