B.Tech in COMPUTER SCIENCE AND ENGINEERING
Course Structure & Syllabus (R-25 Regulations)
Applicable from AY 2025-2026 Batch

I YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	A2001	Matrices and Calculus	3	1	0	4
2	A2002	Engineering Chemistry	3	0	0	3
3	A2006	English for Skill Enhancement	3	0	0	3
4	A2401	Electronic Devices and Circuits	3	0	0	3
5	A2501	Programming for Problem Solving	3	0	0	3
6	A2003	Engineering Chemistry Lab	0	0	2	1
7	A2502	Programming for Problem Solving Lab	0	0	2	1
8	A2301	Engineering Workshop	0	0	2	1
9	A2007	English Language and Communication Skills Lab	0	0	2	1
10		Induction Program				
		Total Credits	**15**	**1**	**8**	**20**

I YEAR II SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	A2008	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2	A2004	Advanced Engineering Physics	3	0	0	3
3	A2302	Engineering Drawing and Computer Aided Drafting	2	0	2	3
4	A2403	Basic Electrical Engineering	3	0	0	3
5	A2504	Data Structures	3	0	0	3
6	A2005	Advanced Engineering Physics Lab	0	0	2	1
7	A2505	Data Structures Lab	0	0	2	1
8	A2404	Basic Electrical Engineering Lab	0	0	2	1
9	A2503	IT Workshop	0	0	2	1
		Total Credits	**14**	**1**	**10**	**20**

II YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	A2510	Discrete Mathematics	3	0	0	3
2	A2511	Computer Organization and Architecture	3	0	0	3
3	A2512	Object Oriented Programming through Java	3	0	0	3
4	A2514	Software Engineering	3	0	0	3
5	A2515	Database Management System	3	0	0	3
6	A2012	Computational Mathematics Lab	0	0	2	1
7	A2513	Object Oriented Programming through Java Lab	0	0	2	1
8	A2516	Database Management Systems Lab	0	0	2	1
9	A2517	Data Visualization- Tableau	0	0	2	1
10	A2013	Environmental Science	1	0	0	1
		Total Credits	**16**	**0**	**8**	**20**

II YEAR II SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1	A2014	Computer Oriented Statistical Methods	3	0	0	3
2	A2518	Operating Systems	3	0	0	3
3	A2506	Python Programming	3	0	0	3
4	A2520	Computer Networks	3	0	0	3
5	A2522	Web Technologies	3	0	0	3
6	A2015	Innovation and Entrepreneurship	2	0	0	2
7	A2519	Operating Systems Lab	0	0	2	1
8	A2521	Computer Networks Lab	0	0	2	1
9	A2507	Python Programming Lab	0	0	2	1
10	A2523	Web Technologies Lab	0	0	2	1
11	A2524	UI Design – Flutter	0	0	2	1
		Total Credits	**17**	**0**	**10**	**22**

Note: Students who wish to exit after **II Year II Semester** has to register for this optional course and acquire the credits allotted by doing 6 weeks work-based vocational course /Internship or Apprenticeship. Please refer **R25 regulations** for more information.

.

III YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1		Automata Theory and Compiler Design	3	0	0	3
2		Machine Learning	3	0	0	3
3		Algorithm Design and Analysis	3	0	0	3
4		Professional Elective-I	3	0	0	3
5		Open Elective-I	2	0	0	2
6		Machine Learning Lab	0	0	2	1
7		Professional Elective-I Lab	0	0	2	1
8		Algorithm Design and Analysis Lab	0	0	2	1
9		Field-Based Research Project	0	0	4	2
10		Node JS/React JS/Django	0	0	2	1
11		Indian Knowledge System	1	0	0	1
		Total Credits	**15**	**0**	**12**	**21**

III YEAR II SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1		Cryptography and Networks Security	3	0	0	3
2		Deep Learning	3	0	0	3
3		Business Economics and Financial Analysis	3	0	0	3
4		Professional Elective-II	3	0	0	3
5		Open Elective–II	2	0	0	2
6		Cryptography and Networks Security Lab	0	0	2	1
7		Deep Learning Lab	0	0	2	1
8		Advanced Data Structures using Python Lab	0	0	2	1
9		Advanced English Communication Skills Laboratory	0	0	2	1
10		Prompt Engineering	0	0	2	1
11		Gender Sensitization Lab*/Human Values and Professional Ethics*	1	0	0	1
		Total Credits	**15**	**0**	**10**	**20**

Note: For the courses Gender Sensitization Lab and Human Values and Professional Ethics – one hour of instruction will be conducted on alternate weeks. For example, if a one-hour class for Gender Sensitization Lab is conducted this week, then a one-hour class for Human Values and Professional Ethics will be conducted in the following week**.**

IV YEAR I SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1		Natural Language Processing	3	0	0	3
2		Cyber Security	3	0	0	3
3		Fundamentals of Management	3	0	0	3
4		Professional Elective-III	3	0	0	3
5		Professional Elective–IV	3	0	0	3
6		Open Elective–III	2	0	0	2
7		Natural Language Processing Lab	0	0	2	1
8		Cyber Security Lab	0	0	2	1
9		Industry Oriented Mini Project/Internship	0	0	4	2
		Total Credits	**17**	**0**	**8**	**21**

IV YEAR II SEMESTER

S.No.	Course Code	Course Title	L	T	P	Credits
1		Professional Elective–V	3	0	0	3
2		Professional Elective–VI	3	0	0	3
3		Project Work	0	0	28	14
		Total Credits	**6**	**0**	**28**	**20**

Professional Electives

Professional Elective-I

Course Code	Course Title
	Artificial Intelligence
	Introduction to Data Science
	Software Testing Methodologies
	Data Mining
	Devops
	Full Stack Development

Professional Elective Lab-I

Course Code	Course Title
	Artificial Intelligence using Python Lab
	Introduction to Data Science Lab
	Software Testing Methodologies Lab
	Data Mining Lab
	Devops Lab
	Full Stack Development Lab

Professional Elective-II

Course Code	Course Title
	Image Processing
	Mining Massive Datasets
	Software Project Management
	Block chain Technology
	Distributed Systems
	Generative AI

Professional Elective-III

Course Code	Course Title
	Computer Vision
	Data Stream Mining
	Information Retrieval Systems
	Scripting Languages
	Cloud Computing
	Vulnerability and Penetration Testing

Professional Elective-IV

Course Code	Course Title
	Augmented Reality & Virtual Reality
	Big Data Analytics
	Agile Methodology
	Quantum Computing
	Robotic Process Automation
	Cyber Forensics

Professional Elective-V

Course Code	Course Title
	Game Theory
	Social Media Mining
	Internet of Things
	Nature Inspired Computing
	Mobile Application Development
	Human Computer Interaction

Professional Elective-VI

Course Code	Course Title
	High Performance Computing
	Distributed Databases
	Graph Theory
	Adhoc and Sensor Networks
	Sustainable Engineering
	Edge Computing

Open Electives

Open Elective-I

Course Code	Course Title
	Operating Systems
	Database Management Systems

Open Elective-II

Course Code	Course Title
	Introduction to Computer Networks
	Software Engineering

Open Elective-III

Course Code	Course Title
	Algorithms Design and Analysis
	Fundamentals of Cyber Security

A2001: MATRICES AND CALCULUS

B.Tech. I Year I Sem. **L T P C**
3 1 0 4

Pre-requisite: Mathematical Knowledge at pre-university level

Objectives: To learn

1. Applying basic operations on matrices and their properties.
2. Concept of a rank of the matrix and applying this concept to know the consistency and solving the system of linear equations.
3. Concept of Eigen values and Eigen Vectors and to reduce the quadratic form to canonical form
4. Geometrical approach to the mean value theorems and their application to the mathematical problems.
5. Finding maxima and minima of functions of two and three variables.
6. Evaluation of multiple integrals and their applications

Course Outcomes: After learning the contents of this paper, the student must be able to

1. Write the matrix representation of a set of linear equations and to analyze the solution of the system of equations
2. Find the Eigen values and Eigen vectors
3. Reduce the Quadratic form to canonical form using orthogonal transformations
4. Solve the applications of the mean value theorems
5. Find the extreme values of functions of two variables with/without constraints
6. Evaluate the multiple integrals and apply the concept to find areas and volumes

UNIT-1: Matrices

Rank of a matrix by Echelon form and Normal form, Inverse of Non-singular matrices by Gauss-Jordan method. System of linear equations: Solving system of Homogeneous and Non-Homogeneous Equations. Gauss Jacobi iteration method, Gauss seide Iteration methods.

UNIT-II: Eigen values and Eigen vectors

Linear Transformation and Orthogonal Transformation: Eigen vectors and their properties, Diagonalization of a matrix, cayleyHamilton Theorem (without proof) Finding inverse and power of a matrix by Cayley-Hamilton Theorem. Quadratic

forms and Nature of the Quadratic Forms Reduction of Quadratic form to canonical form by Orthogonal Transformation.

UNIT-III: Single Variables Calculus

Limit and Continuous of Functions and its properties. Mean value theorems: Rolle's Theorem –Lagrange's Mean value theorem with their Geometrical Interpretation and applications – Cauchy's Mean value Theorem – Taylor's series (All the theorems without proof). Maclaurin's Series.

UNIT-IV: Multivariable Calculus (Partial Differentiation and applications)

Definitions of Limit and continuity, Partial Differentiation: Euler's Theorem, Total derivative, Jaqcobian –Functional dependency & independence, Applications: Maxima and minima of functions of two variables and three variables using method of Lagrange multipliers**.**

UNIT-V: Multivariable Calculus (Integration)

Evaluation of Double Integrals (Cartesian and polar coordinates), change of order of integration(only Cartesian form),Change of variables for double integrals(Cartesian to polar).Evaluation of Triple integrals.
Applications: Areas by double integrals and volumes by triple integrals.

TEXT BOOKS:

1. B.S.Grewal, Higher Engineering Mathematics, Hanna Publishers, 36thedition, 2010.
2. R.K.Jain and S.R.K. Iyenger Advanced Engineering Mathematics, Narosa Publications, 5th Edition, 2016.

REFERENCE BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley&sons, 2006.
2. G.B. Thomas and R.L.Finney, calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
3. N.P.Bali and Manish Goyal.A textbook of Engineering Mathematics, Laxmi Publications, Reprint, 2008.
4. H.K.Dass and Er. Rajnish Verma, Higher Engineering Mathematics, S Chand and Company Limited.

A2002: ENGINEERING CHEMISTRY

B.Tech. I Year I Sem. **L T P C**
3 0 0 3

Pre-requisite: Pre-university Knowledge

Course Objectives:

1. To develop adaptability to new advances in Engineering Chemistry and acquire the essential skills to become a competent engineering professional.
2. To understand the industrial significance of water treatment, fundamental principles of battery chemistry, and the impact of corrosion along with its control methods for structural protection.
3. To impart foundational knowledge of various energy sources and their practical applications in engineering.
4. To equip students with an understanding of smart materials, biosensors, and analytical techniques applicable in engineering, industrial, environmental, and biomedical fields.

Course Outcomes:

1. To analyze the quality of water samples and their fitness for steam generation in industries and suitability for human consumption.
2. To understand the various causes, effects and control techniques of corrosion.
3. To categorize the different types of batteries and fuel cells for their advanced applications and to calculate the calorific values and learn usage of different fuels.
4. To compare the general properties and applications of different polymeric materials like plastics, rubber, conducting polymers and biodegradable polymers.
5. To illustrate various applications of advanced functional materials like smart materials, biosensors and interpretative UV-VIS, IR spectroscopy.

UNIT–I: Water and its treatment

Introduction- Hardness, types, degree of hardness and units. Estimation of hardness of water by complexometric method - Numerical problems. Potable water and its specifications (WHO) - Steps involved in the treatment of potable water - Disinfection of

potable water by chlorination and breakpoint chlorination. Defluoridation - Nalgonda technique.

Boiler troubles: Scales, Sludges and Caustic embrittlement. Internal treatment of boiler feed water - Calgon conditioning, Phosphate conditioning, Colloidal conditioning. External treatment methods - Softening of water by ion-exchange processes. Desalination of brackish water – Reverse osmosis.

UNIT–II: Electrochemistry and Corrosion

Introduction- Electrode potential, standard electrode potential, Nernst equation (no derivation), electrochemical cell - Galvanic cell, cell representation, EMF of cell - Numerical problems. Types of electrodes, reference electrodes - Primary reference electrode - Standard Hydrogen Electrode (SHE), Secondary reference electrode - Calomel electrode.

Corrosion: Introduction- Definition, causes and effects of corrosion – Theories of corrosion, chemical and electrochemical theories of corrosion, Types of corrosion: galvanic, water-line and pitting corrosion. Factors affecting rate of corrosion - Nature of the metal, Nature of the corroding environment. Corrosion control methods - Cathodic protection Methods - Sacrificial anode and impressed current methods.

UNIT–III: Energy sources

Batteries: Introduction – Classification of batteries - Primary, secondary and reserve batteries with examples. Construction, working and applications of Zn-air and Lithium ion battery. Fuel Cells – Differences between a battery and a fuel cell, Construction and applications of Direct Methanol Fuel Cell (DMFC).

Fuels: Introduction and characteristics of a good fuel, Calorific value – Units - HCV, LCV- Dulongs formula - Numerical problems.

Fossil fuels: Introduction, Classification, Petroleum - Refining of Crude oil, Cracking - Types of cracking- Moving bed catalytic cracking. LPG and CNG composition and uses.

Synthetic Fuels: Fischer-Tropsch process, Introduction and applications of Hythane and Green Hydrogen.

UNIT–IV: Polymers

Definition - Classification of polymers: Based on origin and tacticity with examples – Types of polymerization - Addition (free radical addition mechanism) and condensation polymerization. Plastics, Elastomers and Fibers: Definition and applications (PVC, Buna-S, Nylon-6,6). Differences between thermoplastics and thermo setting plastics.

Conducting polymers: Definition and Classification with examples - Mechanism of conduction in transpoly-acetylene and applications of conducting polymers.

Biodegradable polymers: Poly lactic acid and its applications.

UNIT–V: Advanced Functional Materials

Smart materials: Introduction, Classification with examples - Shape Memory Alloys – Nitinol, Piezoelectric materials – quartz and their engineering applications.

Biosensor - Definition, Amperometric Glucose monitor sensor.

Interpretative spectroscopic applications of UV-Visible spectroscopy for Analysis of pollutants in dye industry, IR spectroscopy in night vision-security, Pollution Monitoring-CO sensor (Passive Infrared detection).

TEXT BOOKS:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010.
2. Engineering Chemistry by Rama Devi, Dr. P. Aparna and Rath, Cengage learning, 2025.

REFERENCE BOOKS:

1. Engineering Chemistry: by Thirumala Chary Laxminarayana & Shashikala, Pearson Publications (2020).
2. Engineering Chemistry by Shashi Chawla, Dhanpatrai and Company (P) Ltd., Delhi 2011.
3. Engineering Chemistry by Shikha Agarwal, Cambridge University Press, Delhi 2015.
4. Engineering Analysis of Smart Material Systems by Donald J. Leo, Wiley, 2007.
5. Challenges and Opportunities in Green Hydrogen by Editors: Paramvir Singh, Avinash Kumar Agarwal, Anupma Thakur, R.K Sinha.

USEFUL LINKS:

1. E-content- https://doi.org/10.1142/13094 | October 2023
2. E-book:

   https://archive.org/details/EngineeringChemistryByShashiChawla/page/n11/mode/2u

A2006: ENGLISH FOR SKILL ENHANCEMENT

B.Tech. I Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

1. Enhance vocabulary for effective communication
2. Apply suitable sentence structures in both oral and written communication.
3. Strengthen their reading and study skills.
4. Enable students to compose paragraphs, essays, precis, and formal letters
5. Develop proficiency in writing technical reports.

Course Outcomes: Students will be able to

1. Use appropriate vocabulary in their oral and written communication. (SDG 4,8)
2. Apply the rules of functional grammar and sentence structures effectively. (SDG4)
3. Develop comprehension skills from known and unknown passages (SDG 4,8)
4. Compose paragraphs, essays, precis and draft letters (SDG 4)
5. Prepare well-structured abstracts and reports in various contexts (SDG 4,8)

UNIT-I:

Theme: Perspectives

Lesson on 'The Generation Gap' by Benjamin M. Spock from the prescribed textbook titled English for the Young in the Digital World published by Orient Black Swan Pvt. Ltd.

Vocabulary: The Concept of Word Formation - The Use of Prefixes and Suffixes - Words Often Misspelt - Synonyms and Antonyms.

Grammar: Identifying Common Errors in Writing with Reference to Parts of Speech particularly Articles and Prepositions – Degrees of Comparison.

Reading: Reading and Its Importance - Sub Skills of Reading – Skimming and Scanning.

Writing: Sentence Structures and Types - Use of Phrases and Clauses in Sentences - Importance of Proper Punctuation - Techniques for Writing Precisely – Nature and Style of Formal Writing.

UNIT-II:

Theme: Digital Transformation

Lesson on 'Emerging Technologies' from the prescribed textbook titled English for the Young in the Digital World published by Orient Black Swan Pvt. Ltd.

Vocabulary: Homophones, Homonyms and Homographs.

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

Reading: Reading Strategies - Guessing Meaning from Context – Identifying Main Ideas – Exercises for Practice.

Writing: Paragraph Writing – Types, Structures and Features of a Paragraph - Creating Coherence – Linkers and Connectives - Organizing Principles in a Paragraph – Defining - Describing People, Objects, Places and Events – Classifying - Providing Examples or Evidence - Essay Writing - Writing Introduction and Conclusion.

UNIT-III:

Theme: Attitude and Gratitude

Poems on 'Leisure' by William Henry Davies and 'Be Thankful' - Unknown Author from the prescribed textbook titled English for the Young in the Digital World published by Orient Black Swan Pvt. Ltd.

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 – Identifying Topic Sentence and Providing Supporting Ideas - Exercises for Practice.

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

UNIT-IV:

Theme: Entrepreneurship

Lesson on 'Why a Start-Up Needs to Find its Customers First' by Pranav Jain from the prescribed textbook titled English for the Young in the Digital World published by Orient Black Swan Pvt. Ltd.

Vocabulary: Standard Abbreviations in English – Inferring Meanings of Words through Context – Phrasal Verbs – Idioms.

Grammar: Redundancies and Clichés in Written Communication – Converting Passive to Active Voice and Vice-Versa.

Reading: Prompt Engineering Techniques – Comprehending and Generating Appropriate Prompts - Exercises for Practice.

Writing: Writing Practices - Note Making - Précis Writing.

UNIT-V:

Theme: Integrity and Professionalism

Lesson on 'Professional Ethics' from the prescribed textbook titled English for the Young in the Digital World published by Orient Black Swan Pvt. Ltd.

Vocabulary: Technical Vocabulary and their Usage – One Word Substitutes – Collocations.

Grammar: Direct and Indirect Speech - Common Errors in English (Covering all other aspects of grammar which were not covered in previous units).

Reading: Survey, Question, Read, Recite and Review (SQ3R Method) – Inferring the Meaning and Evaluating a Text - Exercises for Practice.

Writing: Report Writing - Technical Reports - Introduction – Characteristics of a Report – Categories of Reports Formats - Structure of Reports (Manuscript Format) - Types of Reports - Writing a Technical Report.

TEXT BOOKS:

1. Board of Editors. 2025, *English for the Young in the Digital World*, Orient Black Swan Pvt. Ltd.

REFERENCE BOOKS:

1. Swan, Michael. (2016), *Practical English Usage*, Oxford University Press. New Edition.
2. Karal, Rajeevan. 2023, *English Grammar Just for You*, Oxford University Press. New Delhi.
3. 2024, *Empowering with Language: Communicative English for Undergraduates*, Cengage Learning India Pvt. Ltd. New Delhi.
4. Sanjay Kumar & Pushp Lata. 2022, *Communication Skills – A Workbook*, Oxford University Press. New Delhi.
5. Wood, F.T. (2007), *Remedial English Grammar*, Macmillan.
6. Vishwamohan, Aysha. (2013), *English for Technical Communication for Engineering Students*, McGraw-Hill Education India Pvt. Ltd.

USEFUL LINKS:

1. https://hostnezt.com/cssfiles/english/Practical%20English%20Usage%20by%20Michael%20Swan.pdf
2. https://mis.alagappauniversity.ac.in/siteAdmin/dde-admin/uploads/6/__UG_B.A._English_112%2064_Remedial%20English%20Grammar_4066.pdf
3. https://aissmschmct.in/wp-content/uploads/2020/07/BOOK-BSc-HS-Sem-III-HS308-Communication-Skills-1.pdf
4. https://ebooks.lpude.in/new-scheme/arts/ma_english/sem_1/DEENG539_ACADEMIC_ENGLISH.pdf

A2401: ELECTRONIC DEVICES AND CIRCUITS

B.Tech. I Year I Sem. **L T P C**
3 0 0 3

Pre-requisite: 10+2 Physics

Course Objectives:

1. Introduce the fundamental characteristics and applications of semiconductor diodes and their circuit implementations.
2. Explain the working principles, configurations, and characteristics of Bipolar Junction Transistors (BJTs).
3. Develop an understanding of BJT biasing techniques, operating point stability, and thermal considerations.
4. Analyze transistor amplifiers using small-signal models and h-parameter equivalent circuits.
5. Familiarize students with special-purpose diodes and their applications in electronic circuits.
6. Explore the structure, operation, and characteristics of Field Effect Transistors (FETs), MOSFETs, and advanced nanoscale devices like FinFETs and CNTFETs.

Course Outcomes:

1. Explain the I–V characteristics of semiconductor diodes and their use in rectifiers, clippers, clampers, and voltage regulation.
2. Analyze the operation and characteristics of BJTs in different configurations and extract h-parameters.
3. Apply various BJT biasing techniques to establish a stable operating point and mitigate thermal runaway.
4. Evaluate the performance of transistor amplifiers (CE, CB, CC) using small-signal models and h-parameters.
5. Demonstrate the principle and applications of special-purpose diodes such as SCR, tunnel diode, varactor, photodiode, LED, and solar cell.
6. Compare and Contrast FET, MOSFET, FinFET, and CNTFET devices in terms of structure, characteristics, and scaling advantages for modern VLSI.

UNIT-I: Diode Characteristics and Applications

PN junction diode – I-V characteristics, Diode resistance and capacitance, Diode models (Ideal, Simplified, Piecewise Linear), Rectifiers – Half-wave, Full-wave (Center-tap and bridge), Capacitor filter for rectifiers, Clippers and clampers, Zener diode – I-V characteristics and voltage regulation.

UNIT-II: Bipolar Junction Transistor (BJT)

Structure and working principle of BJT, Current components and transistor action, Configurations: Common Base (CB), Common Emitter (CE), Common Collector (CC), Input and output characteristics, Determination of h-parameters from transistor characteristics.

UNIT-III: BJT Biasing

Need for biasing and stabilization, Load line and operating point, Biasing techniques: Fixed bias, Collector-to-base bias, Voltage divider bias, Stability factors and thermal runaway.

UNIT-IV: Transistor Amplifiers

Transistor as a small-signal amplifier, h-parameter equivalent circuit, CE, CB, CC amplifier analysis using h-parameters, Approximate CE model – with and without emitter bypass capacitor.

UNIT-V: Special Devices

Special Purpose Diodes: Principle of Operation of – SCR, Tunnel Diode, Varactor Diode, Photo Diode, Solar Cell, LED and Schottky Diode.

Field Effect Transistors and Advanced Devices: JFET: Structure, operation, and characteristics.

MOSFET: Enhancement and Depletion modes – Structure, operation, and characteristics.

Advanced Devices: FinFETs - 3D structure, Scaling advantages, CNTFETs - Structure, ballistic transport, fabrication, Comparison: CMOS vs. FinFET vs. CNTFET.

TEXT BOOKS:

1. Millman, Jacob, and Christos C. Halkias. Electronic Devices and Circuits. Tata McGraw-Hill, 1991.
2. Boylestad, Robert L., and Louis Nashelsky. Electronic Devices and Circuit Theory. Pearson, 11th ed., 2013.
3. Sedra, Adel S., and Kenneth C. Smith. Microelectronic Circuits. Oxford University Press, 7th ed., 2014.

REFERENCE BOOKS:

1. Bell, David A. Electronic Devices and Circuits. Oxford University Press, 5th ed., 2008.
2. Neamen, Donald A. Electronic Circuit Analysis and Design. McGraw-Hill, 2nd ed., 2001.
3. Salivahanan, S., and N. Suresh Kumar. Electronic Devices and Circuits. McGraw-Hill Education, 4th ed., 2017.
4. Razavi, Behzad. Fundamentals of Microelectronics. Wiley, 2nd ed., 2013.
5. Taur, Yuan, and Tak H. Ning. Fundamentals of Modern VLSI Devices. Cambridge University Press, 2nd ed., 2009.

A2501: PROGRAMMING FOR PROBLEM SOLVING

B.Tech. I Year I Sem. **L T P C**
3 0 0 3

Pre-requisite: Mathematics Knowledge at pre-university level

Course Objectives: The course aims to

1. Understanding problem-solving methodologies using algorithms and flowcharts.
2. Learning fundamentals of C programming language constructs.
3. Implementing control structures, functions and modular programming techniques.
4. Using arrays, strings, structures, pointers, and file handling to solve computational problems.
5. Using pre-processor directives and user-defined data types to enhance code modularity and maintainability.

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

1. Design algorithms and flowcharts to solve problems and Write C programs using appropriate data types, operators, and expressions.
2. Use control flow constructs such as conditional and loop statements to implement logic.
3. Manipulate arrays, strings, and structures to solve real-world problems.
4. Apply modular programming using functions and recursion effectively.
5. Demonstrate effective use of pointers and dynamic memory allocation.
6. Perform file operations and use preprocessor directives in C programs.

UNIT-I: Introduction to Programming

Introduction to Problem Solving: Algorithm, Flowchart, Pseudo code with examples. **Introduction to C**: Structure of a C Program, Compilation and Execution Process, Variables, Constants, Data Types.

Operators: Arithmetic, Relational, Logical, Assignment, Bitwise, Unary, Conditional, Operators precedence and Expression Evaluation, Type Conversion

Input and Output: Formatted/unformatted I/O.

UNIT-II: Control Structures and Arrays

Conditional Branching: if, if-else, nested if-else, if else if ladder, switch-case

Iteration: for, while, do-while

Jump Statements: break, continue, goto.

Arrays: Declaration, Initialization and Access, One Dimensional Arrays and Two Dimensional Arrays, Applications in Problem Solving: Linear Search, Bubble Sort.

UNIT-III: Strings and Functions

String Handling: Declaration, Initialization, Standard Functions: strlen, strcat, strcpy, strcmp, Manual string manipulation (without string.h), array of Strings.

Functions: Declaration, Definition, Calling, Types/Categories of user defined functions, Parameter Passing: Call by Value. Recursion: The Nature of Recursion, Tracing a Recursive Function, Recursive Mathematical Functions, Recursive Functions with Array and String Parameters, Scope and Storage Classes (auto, extern, static, register).

UNIT-IV: Structures, Pointers and Dynamic Memory

Structures and Unions: Declaration, Initialization, Nested Structures, Array of Structures, Passing Structures to Functions, Union.

Pointers: Declaration, Initialization, Pointer Arithmetic, Pointers with Arrays, Functions: Call by Reference, Passing Arrays to Functions, Pointers with Structures.

Dynamic Memory Allocation: malloc, calloc, realloc, free, Self-referential Structure.

UNIT-V: File Handling and Preprocessor Directives

File Handling: Text and Binary Files, File Operations: Opening, reading, writing, closing, File handling functions: fopen, fclose, fgetc, fputc, fgets, fputs, fread, fwrite, fprintf, fscanf. File Positioning: fseek, ftell, rewind

Preprocessor Directives: #define, #include, #ifdef, #ifndef, Command Line Arguments, User-Defined Data Types: enum, typedef.

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B. Koffman, Problem solving and Program Design in C 7th Edition, Pearson.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition).

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill.
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB.
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression).
5. Stephen G. Kochan, Programming in C, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition.
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.

A2003: ENGINEERING CHEMISTRY LAB

B.Tech. I Year I Sem.

L	T	P	C
0	0	2	1

Pre-requisite: Pre-university Knowledge

Course Objectives:

1. Students will understand and perform experiments based on core chemical principles relevant to engineering applications.
2. Students will learn to estimate the hardness of water to assess its suitability for drinking purposes.
3. Students will acquire the ability to perform acid-base titrations using instrumental methods such as conductometry, potentiometry, and pH metry.
4. Students will gain hands-on experience in synthesizing polymers like Bakelite and Nylon – 6, 6 in the laboratory.
5. Students will learn to determine the unknown concentration of potassium permanganate (KMnO4) using a calibration curve.

Course Outcomes:

1. Student is able to determine the hardness of water using complexometric titration.
2. Student is able to determine the corrosion rate on mild steel in different conditions.
3. Student is able to determine the equivalence points of acids and bases using conductometry and pH metry.
4. Student is able to synthesize Polymers like Bakelite and Nylon-6,6.
5. Student is able to determine the concentrations or equivalence points of acids and bases using Potentiometry.

UNIT–I: Volumetric Analysis

Estimation of Hardness of water by EDTA Complexometry method.

UNIT–II: Conductometry

1. Estimation of the concentration of strong acid by Conductometry.
2. Estimation of the concentration of strong and weak acid in an acid mixture by Conductometry.

UNIT–III: Potentiometry

Estimation of concentration of Fe+2 ion by Potentiometry using KMnO4.

UNIT–IV: pH Metry

Determination of an acid concentration using pH meter.

UNIT–V: Preparations

Preparation of Bakelite.

UNIT–VI: Corrosion

Determination of rate of corrosion of mild steel in the presence and absence of inhibitor.

UNIT–VII: Virtual Lab Experiments

1. Construction of Fuel cell and it's working.
2. Smart materials for Biomedical applications.
3. Batteries for electrical vehicles.
4. Functioning of solar cell and its applications.

REFERENCE BOOKS:

1. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, S Chand Publications, New Delhi (2022).

2. Vogel's text book of practical organic chemistry 5th edition.

3. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.

4. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007).

A2502: PROGRAMMING FOR PROBLEM SOLVING LAB

B.Tech. I Year I Sem. **L T P C**
0 0 2 1

Course Objectives: The course aims to

1. Develop proficiency in C programming by understanding syntax, semantics, and program structure.
2. Enhance problem-solving abilities using control structures, arrays, strings, functions, recursion, pointers, and structures.
3. Apply modular programming concepts to implement efficient and reusable code.
4. Utilize dynamic memory allocation, file handling, and command-line arguments for real-world programming scenarios.
5. Strengthen analytical thinking through implementation of numerical, pattern generation, and data processing problems.

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

1. Apply fundamental C programming concepts to develop efficient and modular programs.
2. Analyze, design, and implement C programs involving matrices, file handling, and command-line arguments following standard coding practices.

Simple numeric problems

1) Demonstration Programs on printf( ),scanf( )
2) Write a simple C program that prints the results of all the operators available in C. Read required operand values from standard input
3) Write the program for the simple, compound interest.
4) Ramesh works as a Mess supervisor in an engineering college hostel, where different messes are available based on the years. The student count varies daily in all the hostels due to continuous holidays. Since Ramesh is in charge of the cooking team, he faces difficulty in calculating the quantity of food that needs to be prepared because of the varying student count. Even if a small quantity of food is prepared by the cooking team, it should be divided equally among the number of messes. Ramesh needs an automated software to identify the amount of food available (in the number of packets) and the Mess count. Can you help him divide the food equally and also calculate the remaining quantity of food that will be available after sharing the food equally?
5) Tina's brother gave her a fun mathematical challenge involving a chessboard-like grid. The task is to determine the total number of squares that can be found in an board, where each square is 1cm x 1cm . However, Tina needs to count not only the smallest squares but also the larger squares that can be formed on the board. For example, on a board, there are 9 squares of size 1 x 1 , 4 squares of size 2 x 2 , and 1 square of 3 x3 size , resulting in a total of 14 squares. Your task is to help Tina calculate the total number of squares of all possible sizes on the board.
6) Bhushan is playing chess on an N × M grid, where all the cells start uncolored.

His score starts at zero, and on each turn, he picks an uncolored cell and colors it. The score for that turn is equal to the number of already colored neighboring cells (cells that share a side with it).The game continues until all cells are colored. Bhushan's final score is the total of all the points he earned in each step.He wants to maximize his final score. Can you help him figure out the highest score he can achieve?

7) Surya used to wear a smartwatch when he was on the treadmill and during cycling. Surya's smartwatch displays the total workout time in seconds.But Surya would like to know the time he spent on the workout in H:M:S format.Can you help Surya in knowing the time he spent on the workout in the prescribed format?
8) Flipkart has announced its year-end stock clearance sale, including a contest where users answering the questions can win a Moto One Power for free.
   The task is to display the sum of the first three powers of the given number .
   Pawan is eager to win a Moto One Power, and if you help him solve the task, he may get his favourite mobile. Can you assist him?
9) Write a program for finding the max and min from the three numbers using ternary Operator.
10) Write a C program to evaluate the following expressions.
    a. A+B*C+(D*E) + F*G
    b. A/B*C-B+A*D/3
    c. A++ + ++B - --A
    d. (i++) + (++i)

Programs using control structures

1) Caleb and Irfan are purchasing apples that are priced according to their size, but their budget is limited.To ensure they stay within budget, they plan to choose one small (apple1), one medium(apple2), and one large (apple3) apple.Can you help them determine if their choices fit into their budget? Check the condition: if apple2 is greater than apple1 and apple3 is greater than apple2. If this condition is true, print "Fit into Budget"; otherwise, print "Doesn't fit into Budget.
2) Agathiyan is the Chief In charge of carrying out the World Economic Survey in India. As a part of the survey, his team collected the salaries of the citizens of India. The Salaries of different people are in different number of digits.
   Now Agathiyan would like to classify the earnings of the citizen based on the number of

   digits of his/her salary into 5 different categories as follows:

   Insufficient Earning

   Very Low Earning

   Low Earning

   Sufficient Earning

   High Earning

Can you help him do the above classification if he gives the salary of the particular person to you as input?

3) Write a C program that declares Class awarded for a given percentage of marks, where
mark <40%= Failed, 40% to <60% = Second class, 60% to <70%=First class, >= 70% = Distinction. Read percentage from standard input.

4) Once upon a time in the small village of Codeton, a diligent programmer named Alex was working on a task for the village elders. The elders wanted a program that could take an integer input representing the day of the week and return the corresponding day's name. The challenge was to ensure that the program gracefully handled all possible inputs. Alex, known for their meticulous coding skills, decided to use a switch-case statement for this task. Can you help Alex to complete this task?
5) Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +,-,*, /, % and use Switch Statement).
6) Write a program to display the cube of the numbers up to a given integer
7) Write a C program to find the sum of individual digits of a positive integer
8) Write a C program to test given number is palindrome or not
9) Write a program that prints a multiplication table for a given number and the number of rows in the table. For example, for a number 5 and rows = 3, the output should be:

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
```

10) Jagadeesh is heading to the city of Vizag for work and is in search of a hotel room with a prime number. He is interested in creating an app in C that can determine whether a given room number is prime or not.Can you assist Jagadeesh in developing this C program?
11) A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. A divisor of an integer x is an integer that can divide x evenly.
Given an integer n, print True if n is a perfect number, otherwise print False.

12) Write a C program that prints the first N numbers of the Fibonacci series.
13) Write a program that takes the binary representation of an unsigned integer and prints the number of '1' bits it has (also known as the Hamming weight).
14) Write a C program to generate all the prime numbers between 1 and n.
15) Write a C Program to print the following pattern

1)

```
    *
   * * *
  * * * * *
 * * * * * * *
```

2)

```
    *
   * * *
  * * * * *
 * * * * * * *

  * * * * *

   * * *

    *
```

3)

```
****
***
**
*
```

4)

```
1
22
333
4444
```

5)

```
1
23
456
78910
```

6)

```
A
B C
D E F
G H I J
K L M N O
```

7)

```
* * * * **
*       *
*       *
* * * * **
```

8)

```
8 // n
1
12
1 3
1  4
1   5
1    6
1     7
12345678
```

Programs using Arrays and Strings

1) Write a C program to find the minimum, maximum , sum and average in an array of integers.
2) Write a C program to square the elements that are at even positions and cube the elements that are at odd positions in an array
3) Write a C program to move all the negative elements to one side (i.e., to the left) of the array while maintaining the original order of appearance
4) Write a C program that finds the greatest number from each row of a matrix
5) Write a C program to find addition of two matrices of order m x n ,p x q
6) Write a C program to find multiplication of two matrices of order m x n ,p x q
7) Write a C program to count the number of digits, spaces, special characters, and alphabets in a string
8) Write a C program to count the frequency of each character in a given string.
9) Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)
10) Write a C program to insert a sub-string into a given main string from a given position.
11) Write a C program To delete n Characters from a given position in a given string
12) Write a C program to count the lines, words and characters in a given text.
13) Lokesh usually likes to play cricket, but now, he is bored of playing it too much, so he is trying new games with strings. Lokesh's friend Tina gave him binary strings S and R, each with length N, and told him to make them identical. However, unlike Tina, Lokesh does not have any superpower and Tina lets Lokesh perform only operations of one type: choose any pair of integers (i,j) such that $1 \le i,j \le N$ and swap the i-th and j-th character of S. He may perform any number of operations (including zero). For Lokesh, this is much harder than cricket and he is asking for your help.Tell him whether it is possible to change the string S to the target string R only using operations of the given type.
14) There are N students standing in a row and numbered 1 through N from left to right. You are given a string S with length N, where for each valid i, the i-th character of S is 'g' If the i-th student is a girl or 'b' if this student is a boy. Students standing next to each Other in the row are friends. The students are asked to form pairs for a dance competition. Each pair must consist of a boy and a girl. Two students can only form a pair if they are friends. Each student can only be part of at most one pair. What is the maximum number of pairs that can be formed?
15) Given a string consisting of words and spaces, print the length of the last word in the string. A word is a maximal substring consisting of non-space characters only.
16) You are given an array/list of box sizes sorted in ascending order and an integer representing the size of an item. Your task is to write a code to find the best-fit box size for the given item

17) Given a string , find the first non-repeating character in it and print its index. If it does not exist, print -1
18) Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

Programs using functions and Recursion

1. Write a C program to swap to numbers using call by value.
2. In the coding town of Programland, Jake, a budding coder, has crafted a simple geometry calculator in C. Jake's program features easy-to-use functions that magically calculate the areas of common shapes such as squares, rectangles, triangles, and circles. Each shape has its own dedicated chapter, turning the program into an accessible adventure.John, a student, found joy in using Jake's creation and is now inspired to create his own geometry calculator. Can you help John write the code to build this geometric calculator?
3. Imagine you are a software developer working for an educational institution that needs to automate the process of calculating students' average grades for their performance in three core subjects: Math, English, and Science. The institution wants a program that can efficiently calculate these averages to streamline its grading process.Your task is to design and implement a C program that includes a function called calculateAverage to compute the average grade for a student's scores in these three subjects.
4. Imagine you are working on a payroll management system for a company. As part of the system, you need to develop a C program that calculates the minimum salary among a group of employees. Write a program that prompts the user to input the number of employees and their respective salaries. Utilize an array to store the salaries and to find and display the minimum salary in the dataset.
5. Write a C program to print the transpose of a matrix using functions.
6. Write a C program to calculate x^n using recursion
7. In the fast-paced world of physics research, the "Quantum Dynamics Research Team" led by Dr. Maria Rodriguez is actively exploring groundbreaking experiments. As part of their endeavors, they require a specialized scientific calculator program to compute the sum of a series crucial for their experiments. They've reached out to you to develop this program, and you're determined to meet their needs. Write a C program to help the researchers.
   The series is defined as: S=1 + ½ + 1/3 +……+1/n

8. In the world of computer science education, Professor Anderson is teaching a programming course where students are delving into the fascinating world of algorithms and recursion. As a part of their learning journey, he assigns a task to the students: create a program to print the Fibonacci series up to the nth term. This task aims to reinforce their understanding of recursion and sequence generation. Can you help the students in solving this problem?

9. Write a C program to find the GCD (greatest common divisor) of two given integers.
10. Write a C program to form a string containing all the capital letters found in another string in C using Recursion
11. Write a C program to form a string containing all the capital letters found in another string in C using Recursion

Programs using Pointers and Structures

1. Write a C program to swap two numbers using call by reference
2. Write a program for reading elements using a pointer into an array and display the values using the array
3. Write a program for display values reverse order from an array using a pointer.
4. Mike has a unique way of modifying strings using pointers. He wants to remove all the characters other than alphabets from a string, and then capitalize all the alphabets. Write a C program to help Mike perform this character modification operation using pointers
5. Write a C program to create an one dimensional array dynamically using malloc() to store n elements and find sum, average of array elements
6. You are tasked with creating a program to manage employee information for a small company. Each employee has a unique ID, a name, and a monthly salary. Your program should allow input for multiple employees and then display their details.
7. Develop a program to manage a library's book inventory. The program should collect details for a specified number of books, calculate the total cost of all books, and display details of books whose price exceeds Rs. 500
8. Construct a C program that executes the following operations:
   Establishes a nested structure called Customer, which includes:
   A char array for the customer's name. An int for the customer ID.

   A nested Address structure, containing a char array for the city. Requests the user to input the number of customers to be entered into the system. Initializes an array of Customer structures, sized based on the user's input. Iterates over the array to collect and assign each customer's name, ID, and city to the respective Customer structures. Iterates over the array again to display the detailed information of each customer. This task is to use nested structures to manage and present customer data within your program. Ensure that you prompt the user for all the necessary details and handle the Input and output processes with care.

9. Develop a C program that executes the following operations:
   Introduces a nested structure titled Employee, which should encompass:

   A char array to hold the employee's name.

A nested Date structure, composed of integer fields for the day, month, and year, to record the employee's hire date. Inquires the user for the total count of employees to be cataloged. Generates an array of Employee structures, with its length determined by the user's input. Proceeds through the array to solicit and store each employee's name and hire date within the Employee structures. Traverses the array again to exhibit the name and hire date of each employee.

Programs using Files

1. Write a C Program to read the data from a standard input device, store it in a file and then display the content of a file to standard output device
2. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.
3. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).
4. Emily wants to analyze the contents of a text file. She needs a C program to read the content of the file, count the number of lines, words, and characters, and display these statistics. Implement the program to assist Emily in analyzing the file.
5. We are maintaining a file for employees working in an office of their logged in timings and sign off timings. Write a program that can add the entries of multiple employees every day at the end of the file.
6. Noah needs a C program to read the content of a text file, remove all special characters excluding spaces, and then print the cleaned content. If there are no special characters in the file, he should print the contents as they are. Can you implement the program to assist Noah in achieving this task?
7. Write a C program that takes a numerical input from the user and uses Macros SQUARE and CUBE to calculate either the square or cube of the input, depending on the user's choice. Implement a menu system displaying options for the user to choose between the square and cube operations. Use conditional preprocessor directives to check if the SQUARE and CUBE macros are defined before applying them in the program. If the macros are not defined, prompt an error message
8. Write a C Program to demonstrate command line arguments

TEXT BOOKS:

1. Jeri R. Hanly and Elliot B. Koffman, Problem solving and Program Design in C 7th Edition, Pearson.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition).

REFERENCE BOOKS:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India.
2. E. Balagurusamy, Computer fundamentals and C, 2nd Edition, McGraw-Hill.
3. Yashavant Kanetkar, Let Us C, 18th Edition, BPB.
4. R.G. Dromey, How to solve it by Computer, Pearson (16th Impression).
5. Programming in C, Stephen G. Kochan, Fourth Edition, Pearson Education.
6. Herbert Schildt, C: The Complete Reference, Mc Graw Hill, 4th Edition.
7. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill.

A2301: ENGINEERING WORKSHOP

B.Tech. I Year I Sem. **L T P C**
0 0 2 1

Pre-requisite: Practical Skill

Course Objectives:

1. To introduce students to basic manufacturing processes and workshop practices.
2. To provide hands-on training in carpentry, fitting, welding, sheet metal, and machining.
3. To develop skills in using hand tools and measuring instruments.
4. To enhance safety awareness and proper handling of workshop equipment.
5. To build a foundational understanding of industrial production and fabrication.

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

1. Understand the basic manufacturing processes and operations.
2. Use hand tools and equipment safely and efficiently.
3. Perform basic operations in carpentry, fitting, welding, sheet metal work, and machining.
4. Read and interpret workshop drawings.
5. Develop teamwork, time management, and quality awareness in a workshop environment.

1. **TRADES FOR EXERCISES:**

At least two exercises from each trade:

i. **Carpentry:** T- Lap Joint, Dovetail Joint, Mortise and Tenon Joint.
ii. **Fitting:** V- Fit, Dovetail Fit and Semi-circular fit.
iii. **Tin Smithy:** Square Tin, Rectangular Tray and Conical Funnel.
iv. **Foundry:** Preparation of Green Sand Mould using Single Piece and Split Pattern.
v. **Welding Practice:** Arc Welding and Gas Welding.
vi. **House wiring:** Parallel and Series, Two-way Switch and Tube Light.
vii. **Black Smithy:** Round to Square, Fan Hook and S- Hook.

2. **TRADES FOR DEMONSTRATION AND EXPOSURE:**

Plumbing, Machine Shop, Metal Cutting (Water Plasma), Power tools in construction and Wood Working.

TEXT BOOKS:

1. Workshop Practice, B. L. Juneja, Cengage Learning India, 1st edition, 2015.
2. Workshop Practice Manual, K. Venkata Reddy, BS Publication, 6th Edition, Rpt. 2025.

REFERENCE BOOKS:

1. Workshop Manual, K. Venugopal, Anuradha Publications, 2012th edition, 2012.

A2007: ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

B.Tech. I Year I Sem. **L T P C**
0 0 2 1

Course Objectives:

Listening Skills

1. To enable students develop their active listening skills.
2. To equip students with necessary training in listening so that they can comprehend the speech of people from different linguistic backgrounds.

Speaking Skills

3. To improve pronunciation and neutralize accent.
4. To enable students express themselves fluently and appropriately.
5. To practice speaking in social and professional contexts.

Course Outcomes: Students will be able to

1. Listen actively and identify important information in spoken texts (SDG4)
2. Interpret speech and infer the intention of the speaker. (SDG4)
3. Improve accent for intelligibility. (SDG4,8,10)
4. Speak fluently with clarity and confidence. (SDG4,8,10)
5. Use language in real-life situations. (SDG4,8)

Syllabus: English Language and Communication Skills Lab (ELCS) shall have two parts:

a. **Computer Assisted Language Learning (CALL) Lab which focuses on listening skills.**
b. **Interactive Communication Skills (ICS) Lab which ocuses on speaking skills.**

EXERCISE – I

CALL Lab:

Instruction: Speech Sounds, Listening Skill - Importance, Purpose, Types, Barriers, and Active Listening.

Practice: Listening to Distinguish Speech Sounds (Minimal Pairs) and Testing Exercises.

ICS Lab:

Diagnostic Test: Activity titled ‘Express Your View’.

Instruction: Spoken and Written language, Formal and Informal English, Greetings, Introducing Oneself and Others.

Practice: Any Ice-Breaking Activity**.**

EXERCISE – II

CALL Lab:

Instruction: Listening vs. Hearing- Barriers to Listening

Practice: Listening for General Information- Multiple Choice Questions, Listening Comprehension Exercises (It is essential to identify a suitable passage with exercises for practice)

ICS Lab:
Instruction: Features of Good Conversation- Strategies for Effective Communication
Practice: Role Play Activity-Situational Dialogues- Expressions used in various situations-Making Requests and Seeking Permissions- Taking leave- Telephone Etiquette.

EXERCISE – III

CALL Lab:
Instruction: Errors in Pronunciation- Tips for Neutralizing Mother Tongue Influence (MTI)
Practice: Differences between British and American Pronunciation- Listening Comprehension Exercises.

ICS Lab:
Instruction: Describing Objects, Situations, Places, People, and Events
Practice: Picture Description Activity: Looking at a picture and describing objects, situations, places, people, and events. (A wide range of materials and handouts are to be made available in the lab)

EXERCISE – IV

CALL Lab:
Instruction: Techniques for Effective Listening
Practice: Listening for Specific Details- Listening- Gap Fill Exercises- Listening Comprehension Exercises. (It is essential to identify a suitable passage with exercises for practice)

ICS Lab:
Instruction: How to Tell a Good Story- Story Star- Sequencing- Creativity
Practice: Activity on Telling and Retelling Stories- Collage.

EXERCISE – V

CALL Lab:
Instruction: Identifying the Literal and Implied Meaning
Practice: Listening for Evaluation- Write the Summary- Listening Comprehension Exercises.
(It is essential to identify a suitable passage with exercises for practice)

ICS Lab:
Instruction: Understanding Non-Verbal Communication
Practice: Silent Speech- Dumb Charades Activity
* Post-Assessment Test on 'Express Your View'.

Minimum Requirement of Infrastructural Facilities for ELCS lab

1.Computer Assisted Language Learning (CALL) Lab:
The Computer Assisted Language Learning Lab has to accommodate 40 students with 40 systems, with one Master Console, LAN facility and English language learning software for self-study by students.

System Requirement (Hardware component):
Computer network with LAN facility (minimum 40 systems with multimedia) with the following specifications:

i) Computers with Suitable Configuration
ii) High Fidelity Headphones.

2. Interactive Communication Skills (ICS) Lab:
The Interactive Communication Skills Lab: A Spacious room with movable chairs and audiovisual aids with a Public Address System, a T. V. or LCD, a digital stereo – audio & video system and camcorder etc.

Note: English Language Teachers are requested to prepare Materials / Handouts for each Activity for the Use of those Materials in CALL & ICS Labs.
SUGGESTED SOFTWARE:

1. Cambridge Advanced Learners' English Dictionary with CD.
2. Grammar Made Easy by Darling Kindersley.
3. Punctuation Made Easy by Darling Kindersley.
4. Oxford Advanced Learner's Compass, 10th Edition.
5. English in Mind (Series 1-4), Herbert Puchta and Jeff Stranks with Meredith Levy, Cambridge.
6. English Pronunciation in Use (Elementary, Intermediate, Advanced) Cambridge University Press.
7. English Vocabulary in Use (Elementary, Intermediate, Advanced) Cambridge University Press.
8. TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS).

REFERENCE BOOKS:

1. Shobha, KN & Rayen, J. Lourdes. (2019). Communicative English – A workbook. Cambridge University Press.
2. Board of Editors. (2016). ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities. Orient Black Swan Pvt. Ltd.
3. Mishra, Veerendra et al. (2020). English Language Skills: A Practical Approach. Cambridge University Press.
4. (2022). English Language Communication Skills – Lab Manual cum Workbook. Cengage Learning India Pvt. Ltd.
5. Ur, Penny and Wright, Andrew. 2022. Five Minute Activities – A Resource Book for Language Teachers. Cambridge University Press.

A2008: ORDINARY DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS

B.Tech. I Year II Sem.

L	T	P	C
3	1	0	4

Pre-requisite: Mathematical Knowledge at pre-university level.

Course Objectives: To learn

1. Methods of solving the differential equations of first and higher order.
2. Concepts and properties of Laplace transforms.
3. Solving ordinary differential equations using Laplace transforms techniques.
4. The physical quantities involved in engineering fields related to vector-valued functions.
5. The basic properties of vector-valued functions and their applications to line, surface, and volume integrals.

Course Outcomes: After completion of the course the student must be able to

1. Identify whether a given differential equation of first order is exact or not.
2. Solve higher-order differential equations and apply the concept to real-world problems.
3. Use Laplace Transform techniques for solving Ordinary Differential Equations.
4. Evaluate Line, Surface, and Volume integrals and convert them from one to another.

UNIT – I: First Order Ordinary Differential Equations

Exact differential equations, Equations reducible to exact differential equations, linear and Bernoulli's equations, Orthogonal Trajectories (only in Cartesian Coordinates).
Applications: Newton's law of cooling, Law of natural growth and decay.

UNIT – II: Ordinary Differential Equations of Higher Order

Higher-order 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^k V(x). Method of variation of parameters.

UNIT – III: Laplace Transforms

Laplace Transform of standard functions, first shifting theorem, Laplace transforms of functions multiplied by 't' and divided by 't'. Laplace transforms of derivatives and integrals of functions, Evaluation of integrals by Laplace transforms, Laplace transform of periodic functions. Inverse Laplace transform by different methods, convolution theorem (without proof).
Applications: Solving Initial value problems by Laplace Transform method.

UNIT – IV: Vector Differentiation

Vector point functions and scalar point functions, Gradient, Divergence and Curl, Solenoidal and Irrotational vectors, Directional derivatives, Scalar potential functions, Vector Identities.

UNIT – V: Vector Integration

Line, Surface and Volume Integrals. Theorems of Green, Gauss and Stokes (without proofs) and their applications.

TEXT BOOKS:

1. B.S. Grewal, *Higher Engineering Mathematics*, Khanna Publishers, 36th Edition, 2010.
2. R.K. Jain and S.R.K. Iyengar, *Advanced Engineering Mathematics*, Narosa Publications, 5th Edition, 2016.

REFERENCE BOOKS:

1. Erwin Kreyszig, *Advanced Engineering Mathematics*, 9th Edition, John Wiley & Sons, 2006.
2. G.B. Thomas and R.L. Finney, *Calculus and Analytic Geometry*, 9th Edition, Pearson, Reprint, 2002.
3. N.P. Bali and Manish Goyal, *A Text Book of Engineering Mathematics*, Laxmi Publications, Reprint, 2008.
4. H. K. Dass and Er. Rajnish Verma, *Higher Engineering Mathematics*, S Chand and Company Limited, New Delhi.

A2004: ADVANCED ENGINEERING PHYSICS

B.Tech. I Year II Sem.

L	T	P	C
3	0	0	3

Pre-requisite: 10+2 Physics

Course Objectives:

1. To study crystal structures, defects, and material characterization techniques like XRD and SEM.
2. To understand fundamental concepts of quantum mechanics and their applications in solids and nano-materials.
3. To introduce quantum computing principles, quantum gates, and basic quantum algorithms.
4. To learn the properties and applications of magnetic and dielectric materials.
5. To explore the working and applications of lasers and fiber optics in modern technology.

Course Outcomes:

1. Analyze crystal structures, identify defects, and apply XRD and SEM techniques for material Characterization.
2. Apply quantum mechanical principles to explain particle behavior and energy band formation in solids.
3. Understand quantum computing concepts, use quantum gates and Explain basic quantum algorithms.
4. Classify magnetic and dielectric materials and explain their properties, synthesis, and applications.
5. Explain the principles of lasers and fibre optics and their applications in communication and sensing.

UNIT – I: Crystallography & Material Characterization

Introduction: Unit cell, space lattice, basis, lattice parameters; crystal structures, Bravais lattices, packing factor: SC, BCC, FCC; Miller indices, inter-planar distance; defects in crystals (Qualitative): point defects, line defects, surface defects and volume defects. Concept of nanomaterials –Surface to volume ratio, X -ray diffraction: Bragg's law, powder method, calculation of average crystallite size using Debye-Scherer's formula, scanning electron microscopy (SEM): block diagram, working principle.

UNIT - II: Quantum Mechanics

Introduction, de-Broglie hypothesis, Heisenberg uncertainty principle, physical significance of wave function, postulates of quantum mechanics: operators in quantum mechanics, Eigen values and Eigen functions, expectation value; Schrödinger's time independent wave equation, particle in a 1D box, Bloch's theorem (qualitative), Kronig-Penney model (qualitative): E-k diagram, effective mass of electron, formation of energy bands, origin of band gap, classification of solids, Concept of discrete energy levels and quantum confinement in nano materials.

UNIT - III: Quantum Computing

Introduction, linear algebra for quantum computation, Dirac's Bra and Ket notation and their properties, Hilbert space, Bloch's sphere, concept of quantum computer, classical bits, Qubits, multiple Qubit system, quantum computing system for information processing, development of of quantum systems, quantum measurements, entanglement, quantum gates, challenges and advantages of quantum computing over classical computation, quantum algorithms: Shor, Grover.

UNIT - IV: Magnetic and Dielectric Materials

Introduction to magnetic materials, origin of magnetic moment-classification of magnetic materials, hysteresis, Weiss domain theory of ferromagnetism, soft and hard magnetic materials, synthesis of ferrimagnetic materials using Sol-gel method, Applications: magnetic hyperthermia for cancer treatment, magnets for EV, Giant Magneto Resistance (GMR) device. Introduction to dielectric materials, types of polarization (qualitative): electronic, ionic & orientation; ferroelectric, piezoelectric, pyroelectric materials and their applications: Ferroelectric Random- Access Memory (Fe-RAM), load cell and firecell.

UNIT - V: Laser and Fibre Optics

Introduction to laser, characteristics of laser, Einstein coefficients and their relations, meta stable state, population inversion, pumping, lasing action, Ruby laser, He-Ne laser, CO2 laser, semiconductor diode laser, applications: Bar code scanner, LIDAR for automatic vehicle. Introduction to fibre optics, total internal reflection, construction of optical fibre, acceptance angle, numerical aperture, classification of optical fibres, losses in optical fibre, applications: optical fibre for communication system, sensor for structural health monitoring.

TEXT BOOKS:

1. Walter Borchardt-Ott, *Crystallography: An Introduction*, Springer.
2. Charles Kittel, *Introduction to Solid State Physics*, John Wiley & Sons, Inc.
3. Thomas G. Wong, *Introduction to Classical and Quantum Computing*, Rooted Grove.

REFERENCE BOOKS:

1. Jozef Gruska, *Quantum Computing*, McGraw Hill.
2. Michael A. Nielsen & Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press.
3. John M. Senior, *Optical Fiber Communications Principles and Practice*, Pearson Education Limited.

USEFUL LINKS:

1. https://shijuinpallotti.wordpress.com/wp-content/uploads/2019/07/optical-fiber-communications-principles-and-pr.pdf
2. https://www.geokniga.org/bookfiles/geokniga-crystallography_0.pdf
3. https://dpbck.ac.in/wp-content/uploads/2022/10/Introduction-to-Solid-State-PhysicsCharles- Kittel.pdf
4. https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf
5. https://www.fi.muni.cz/usr/gruska/qbook1.pdf
6. https://profmcruz.wordpress.com/wp-content/uploads/2017/08/quantum-computation-and-quantum-information-nielsen-chuang.pdf

A2302: ENGINEERING DRAWING AND COMPUTER AIDED DRAFTING

B.Tech. I Year II Sem.

L	T	P	C
2	0	2	3

Course Objectives: To learn

1. To introduce the fundamentals of engineering drawing and projection systems.
2. To develop skills in constructing orthographic, isometric, and sectional views.
3. To train students in interpreting and creating technical drawings using CAD tools.
4. To familiarize students with dimensioning standards and drafting conventions.
5. To bridge manual drafting techniques with computer-aided drafting practices.

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

1. Understand and apply the principles of orthographic and isometric projections.
2. Create sectional views and dimensioned drawings using BIS standards.
3. Use CAD software to generate 2D engineering drawings.
4. Visualize and construct solid models from 2D views.
5. Interpret and produce engineering drawings of mechanical components and assemblies.
6. Demonstrate drafting skills for practical and industrial applications.

UNIT-I: Introduction to Engineering Graphics (Conventional)
Principles of Engineering Graphics and their Significance, Geometrical Constructions, Scales, Plain and Diagonal, Conic Sections including the Rectangular Hyperbola, General method only. Cycloid, Epicycloid and Hypocycloid.

UNIT-II: Orthographic Projections (Conventional and Computer Aided)
Principles of Orthographic Projections, Conventions, Projections of Points and Lines, Projections of Plane regular geometric figures. Auxiliary Planes. Computer aided orthographic projections, points, lines and planes. Introduction to Computer aided drafting, views, commands and conics.

UNIT-III: Projections of Regular Solids (Conventional and Computer Aided)
Auxiliary Views, Sections or Sectional views of Right Regular Solids, Prism, Cylinder, Pyramid, Cone, Auxiliary views, Computer aided projections of solids, sectional views.

UNIT-IV: Development of Surfaces (Conventional)
Prism, Cylinder, Pyramid and Cone.

UNIT-V: Isometric Projections (Conventional and Computer Aided)
Principles of Isometric Projection, Isometric Scale, Isometric Views, Conventions, Isometric Views of Lines, Plane Figures, Simple and Compound Solids, Isometric Projection of objects having non, isometric lines. Isometric Projection of Spherical Parts.

Conversion of Isometric Views to Orthographic Views and Vice- versa, Conventions. Conversion of orthographic projection into isometric view.

Note:

1. The End Semester Examination will be in conventional mode.
2. CIE – I will be in conventional mode.
3. CIE – II will be using Computer.

TEXT BOOKS:

1. Engineering Drawing, N. D. Bhatt, Charotar, 54th Edition, 2023.
2. Engineering Drawing and graphics Using AutoCAD, T. Jeyapoovan and Vikas, S. Chand and company Ltd., 3rdEdition, 2010.

REFERENCE BOOKS:

1. Engineering Drawing, Basant Agrawal and C.M. Agrawal, McGraw Hill, 3rd Edition, 2019.
2. Engineering Graphics and Design, WILEY, John Wiley and Sons Inc, 3rdEdition, 2020.
3. Engineering Drawing, M. B. Shah and B.C. Rane, Pearson, 2nd Edition, 2009.
4. Engineering Drawing, N. S. Parthasarathy and Vela Murali, Oxford, 1st Edition, 2015.
5. Computer Aided Engineering Drawing, K. Balaveera Reddy, CBS Publishers, 2nd Edition, 2015.

A2403: BASIC ELECTRICAL ENGINEERING

B.Tech. I Year II Sem. **L T P C**
3 0 0 3

Pre-requisite: Mathematics

Course Objectives:

1. Introduce the fundamental concepts of electrical circuit elements, DC circuit laws, and basic theorems for circuit analysis.
2. Develop an understanding of AC circuit analysis using phasor representation, power calculations, and three-phase systems.
3. Explain the principles, construction, and performance evaluation of transformers.
4. Familiarize students with the construction, operation, and characteristics of DC machines, induction motors, and synchronous generators.
5. Provide knowledge of essential electrical installation components such as LT switchgear, cables, earthing, and batteries.
6. Enable students to perform practical calculations for energy consumption, power factor correction, and battery backup for real-world applications.

Course Outcomes:

1. Apply KVL, KCL, and network theorems (Superposition, Thevenin, Norton) to analyze DC circuits and transient behavior of RL/RC circuits.
2. Analyze single-phase and three-phase AC circuits using phasor representation and determine active, reactive, and apparent power along with resonance conditions.
3. Explain the construction, equivalent circuit, losses, and performance characteristics of single-phase and three-phase transformers.
4. Evaluate the operating principles and performance of DC machines, induction motors, and synchronous generators.
5. Demonstrate the applications of LT switchgear, cables, earthing systems, and batteries in electrical installations.
6. Calculate practical electrical parameters such as energy consumption, power factor improvement, and battery backup for given systems.

UNIT-I: D.C. Circuits

Electrical circuit elements (R, L and C), voltage and current sources, KVL&KCL, analysis of simple circuits with dc excitation. Superposition, Thevenin and Norton Theorems. Time-domain analysis of first-order, RL and RC circuits.

UNIT-II: A.C. Circuits

Representation of sinusoidal waveforms, peak and RMS values, phasor representation, real power, reactive power, apparent power, power factor, Analysis of single-phase ac circuits consisting of R, L, C, RL, RC, RLC combinations (series and parallel), resonance in series R-L-C circuit. Three-phase balanced circuits, voltage and current relations in star and delta connections.

UNIT-III: Transformers
Ideal and practical transformer, equivalent circuit, losses in transformers, regulation and efficiency. Auto-transformer and three-phase transformer connections.

UNIT-IV: Electrical Machines
Construction and working principle of dc machine, performance characteristics of dc shunt machine. Generation of rotating magnetic field, Construction and working of a three-phase induction motor, Significance of torque-slip characteristics. Single-phase induction motor, Construction and working. Construction and working of synchronous generator.

UNIT-V: Electrical Installations
Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, ELCB, Types of Wires and Cables, Earthing. Types of Batteries, Important Characteristics for Batteries. Elementary calculations for energy consumption, power factor improvement and battery backup.

TEXT BOOKS:

1. D.P. Kothari and I. J. Nagrath, “Basic Electrical Engineering”, Tata McGraw Hill, 4th Edition, 2019.
2. MS Naidu and S Kamakshaiah, “Basic Electrical Engineering”, Tata McGraw Hill, 2nd Edition, 2008.

REFERENCE BOOKS:

1. P. Ramana, M. Suryakalavathi, G.T. Chandrasheker, “Basic Electrical Engineering”, S. Chand, 2nd Edition, 2019.
2. D. C. Kulshreshtha, “Basic Electrical Engineering”, McGraw Hill, 2009.
3. M. S. Sukhija, T. K. Nagsarkar, “Basic Electrical and Electronics Engineering”, Oxford, 1st Edition, 2012.
4. Abhijit Chakrabarthi, Sudipta Debnath, Chandan Kumar Chanda, “Basic Electrical Engineering”, 2nd Edition, McGraw Hill, 2021.
5. L. S. Bobrow, “Fundamentals of Electrical Engineering”, Oxford University Press, 2011.
6. E. Hughes, “Electrical and Electronics Technology”, Pearson, 2010.
7. V. D. Toro, “Electrical Engineering Fundamentals”, Prentice Hall India, 1989.

A2504: DATA STRUCTUFRES

B.Tech. I Year II Sem. **L T P C**
3 0 0 3

Pre-requisite: Programming for Problem Solving

Course Objectives: The course aims to

1. Understand fundamental concepts, classifications, and operations of data structures.
2. Implement and analyze searching and sorting algorithms for efficient data processing.
3. Learn linear and non-linear data structures and their applications.
4. Apply suitable data structures for problem-solving and algorithm optimization.
5. Gain proficiency in advanced concepts like balanced search trees, hashing, and collision resolution.

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

1. Select appropriate searching and sorting techniques for given application.
2. Understand Linear data structures operations.
3. Perform Various Linked List operations and its types.
4. Implement Linear data structures Applications.
5. Solve the given Problem using Non Linear data structures.
6. Apply the concepts of Graph Traversals and Hashing for efficient Searching.

UNIT-I:
INTRODUCTION TO DATA STRUCTURES: Introduction, Classification of Data Structures, Operations on Data Structures.
SEARCHING AND SORTING: Jump search, binary search, interpolation search, selection sort, Insertion sort, shell sort, Radix sort.

UNIT-II:
STACKS: Introduction, Array Representation of Stack, Operations on Stack. **APPLICATIONS OF STACKS:** Infix-to-Postfix conversion, evaluating Postfix expressions.
QUEUES: Introduction, Array representation of Queue, Operations on a Queue, Circular Queue, Operations on a Circular Queue, Double Ended Queue, Operations on Double Ended Queue.

UNIT-III:
LINKED LISTS: Introduction, Singly Linked List: Representation of a Singly Linked List, Operations on a Single Linked List (Create, insert, delete, reverse, traverse and count), and Operations on a Double Linked list (Create, insert, delete, reverse, traverse and count), Circular Linked Lists.
APPLICATIONS OF LINKED LISTS: Implementation of Stack using Linked List, Implementation of Queue using linked list.

UNIT-IV:

TREES: Definition, Basic Terminologies, Binary Trees: Properties, Types, Representation of a Binary Tree using Array and Linked List and Tree Traversals: Pre order, In order and Post order.

HEAPS: Binary Heap.

SEARCH TREES: Binary Search Trees: Definition, Operations: Insertion, searching, find Minimum, find Maximum and Deletion.

BALANCED SEARCH TREES: AVL Trees: Definition, Rotations, Operations – Insertion, deletion and Searching, B Trees, B+ Trees.

UNIT-V:

GRAPHS: Definition, Basic Terminologies and Representation, Graph Traversals-Breadth First Search (BFS) and Depth First Search (DFS).

HASHING: Hash table representation, hash functions, collision resolution Techniques: separate chaining, open addressing-linear probing, quadratic probing, double hashing, and rehashing.

TEXT BOOKS:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.
2. Data Structure using C – Reema Thareja, 3rd Edition, Oxford University Press.

REFERENCE BOOKS:

1. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.
2. Horowitz, Ellis, Sahni, Sartaj, Anderson-Freed, Susan, Fundamentals of Data Structure in C, 2nd Edition, University Press, India.
3. Samanta Debasis, Classic Data Structures, 2nd Edition, Prentice Hall of India.

A2005: ADVANCED ENGINEERING PHYSICS LAB

B.Tech. I Year II Sem. **L T P C**
0 0 2 1

Pre-requisite: 10+2 Physics

Course Objectives:

1. To provide practical exposure to advanced concepts in solid-state and modern physics
2. To synthesize and study the physical properties of materials like semiconductors, ferromagnetic, and ferroelectric substances
3. To perform semiconductor characterization using Hall effect and band gap experiments
4. To explore the working principles of lasers and optical fibers through hands-on experiments
5. To develop skills in data analysis, interpretation, and scientific reporting

Course Outcomes:

1. Synthesize and analyze nanomaterials such as magnetite (Fe_3O_4) using chemical methods.
2. Determine key electrical, magnetic, and optical properties of semiconductors and other functional materials.
3. Characterize semiconductors using Hall effect and energy gap measurement techniques.
4. Demonstrate working knowledge of laser systems and optical fiber parameters through experimental study.
5. Apply scientific methods for accurate data collection, analysis, and technical report writing.

List of Experiments:

1. Synthesis of magnetite (Fe_3O_4) powder using sol-gel method.
2. Determination of energy gap of a semiconductor.
3. Determination of Hall coefficient and carrier concentration of a given semiconductor.
4. Determination of magnetic moment of a bar magnet and horizontal earth magnetic field.
5. Study of B-H curve of a ferro magnetic material.
6. Study of P-E loop of a given ferroelectric crystal.
7. Determination of dielectric constant of a given material.
8. Determination of Curie's temperature of a given ferroelectric material.
9. A. Determination of wavelength of a laser using diffraction grating.
   B. Study of V-I & L-I characteristics of a given laser diode.
10. A. Determination of numerical aperture of a given optical fibre.
    B. Determination of bending losses of a given optical fibre.

Note: Any **8 Experiments** has to be performed.

A2505: DATA STRUCTURES LAB

B.Tech. I Year II Sem.

L	T	P	C
0	0	2	1

Course Objectives: The course aims to

1. Implement basic searching and sorting algorithms.
2. Design linear and non-linear data structures using arrays and linked lists.
3. Solve expression conversion, evaluation, and traversal problems.
4. Apply advanced structures like balanced trees and hashing.
5. Build modular C programs for real-world problem-solving.

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

1. Apply searching, sorting, hashing, and core operations on linear and nonlinear data structures, and analyze their time and space complexities.
2. Evaluate, design, and implement efficient solutions by selecting appropriate data structures for given problems.

List of Programs:

1. Write a Program to implement
   a) Jump search b) Binary search c) Interpolation search
2. Write a Program that implements the following sorting methods to sort a given list of integers in ascending order
   a) Selection sort b) Insertion sort c) Shell Sort d) Radix sort
3. Write a Program that implement stack (its operations) using Arrays.
4. Write a Program that uses Stack to
   a) Convert the infix expression to postfix expression
   b) Evaluate the postfix expression
5. You are building a text editor that needs to ensure the correctness of parentheses in a user's input. Implement a function that checks whether the parentheses in a given string are balanced using the stack data structure. The string must contain the following types of parentheses: (), [], and {}
6. Write a Program that implements the following with its operations using Arrays:
   a) Queue b) Circular Queue
7. You're presented with a queue Q containing N elements, and your objective is to implement a function called rev() that reverses the order of the elements in the queue. How would you approach this task?
8. Write a Program that uses functions to perform the following operations on singly linked list
   a) Creation b) Insertion c) Deletion d) Traversal
9. Write a Program that uses functions to perform the following operations on doubly linked list
   a) Creation b) Insertion c) Deletion d) Traversal
10. Write a Program that uses functions to perform the following operations on Circular linked list

a) Creation b) Insertion c) Deletion d) Traversal

11. Write a Program that implements the following with its operations using Linked List
    a) Stack b) Queue
12. Write a Program to implement the Binary tree traversal methods.
13. Write a Program to implement the following with its operations
    a) Binary Search tree b) AVL tree
14. You are tasked with developing a program to analyze binary tree traversals. In particular, you need to determine if a given preorder traversal sequence represents a skewed Binary Search Tree (BST). A skewed BST is a binary tree where each node has at most one child. Your goal is to implement a C program that takes a sequence of integers representing a preorder traversal of a binary tree and determines whether the corresponding binary tree is skewed or not.
15. Write a Program to implement the graph traversal methods.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, Universities Press.
2. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

REFERENCE BOOKS:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.

A2404: BASIC ELECTRICAL ENGINEERING LAB

B.Tech. I Year II Sem. **L T P C**
0 0 2 1

Pre-requisite: Mathematics

Course Objectives:

1. Provide hands-on experience in verifying basic electrical laws (KVL, KCL) and theorems (Thevenin, Norton, Superposition).
2. Develop practical skills to analyze DC transients and AC resonance phenomena in RLC circuits.
3. Familiarize students with the performance evaluation of single-phase transformers through voltage, current, efficiency, and regulation tests.
4. Demonstrate the characteristics of DC and AC machines, including DC shunt motors, three-phase induction motors, and alternators.
5. Train students in experimental measurement of electrical quantities such as voltage, current, active power, reactive power, and impedance.
6. Strengthen the ability to apply theoretical concepts of Basic Electrical Engineering to real-world systems through laboratory practice.

Course Outcomes:

1. Verify KVL, KCL, and network theorems (Thevenin, Norton, and Superposition) experimentally.
2. Analyze the transient response of RL/RC circuits and resonance behavior in RLC circuits.
3. Determine impedance, current, and power in AC series circuits (RL, RC, RLC) using experimental methods.
4. Evaluate the voltage, current, efficiency, and regulation characteristics of single-phase and three-phase transformers.
5. Examine the performance characteristics of rotating machines such as DC shunt motors, induction motors, and alternators.
6. Measure and Interpret active and reactive power in balanced three-phase systems using practical setups.

PART - A: Compulsory

1. Verification of KVL and KCL
2. Verification of Thevenin's and Norton's theorem
3. Transient Response of Series RL and RC circuits for DC excitation
4. Resonance in series RLC circuit
5. Calculations and Verification of Impedance and Current of RL, RC and RLC series circuits
6. Measurement of Voltage, Current and Real Power in primary and Secondary Circuits of a Single-Phase Transformer
7. Performance Characteristics of a DC Shunt Motor
8. Torque-Speed Characteristics of a Three-phase Induction Motor

PART - B: Minimum Two experiments from the given list

1. Verification of Superposition theorem
2. Three Phase Transformer: Verification of Relationship between Voltages and Currents (Star-Delta, Delta-Delta, Delta-star, Star-Star)
3. Load Test on Single Phase Transformer (Calculate Efficiency and Regulation)
4. Measurement of Active and Reactive Power in a balanced Three-phase circuit
5. No-Load Characteristics of a Three-phase Alternator

A2503: IT WORKSHOP

B.Tech. I Year II Sem.

L T P C
0 0 2 1

Course Objectives:

1.The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, PowerPoint and Publisher.

Course Outcomes:

1. Perform Hardware troubleshooting
2. Understand Hardware components and inter dependencies
3. Safeguard computer systems from viruses/worms
4. Document/ Presentation preparation
5. Perform calculations using spreadsheet

PC Hardware

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both Windows and Linux. Lab instructors should verify the work and follow it up with a Viva.

Internet & World Wide Web

Task 1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW and LAN.

Task 2: Web Browsers, Surfing the Web: Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

Task 3: Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to

the students for which they need to search on Google. This should be demonstrated to the instructors by the student.
Task 4: Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

LaTeX and WORD
Task 1 – Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word. Importance of LaTeX and MS office or equivalent (FOSS) tool as word Processors. Details of the four tasks and features that would be covered in each. Using LaTeX and word – Accessing, overview of toolbars, saving files, using help and resources, rulers, format painter in word.
Task 2: Using LaTeX and Word to create a project certificate. Features to be covered: Formatting Fonts in word , Drop Cap in word , applying Text effects, using Character Spacing, Borders and Colors, inserting Header and Footer, using Date and Time option in both LaTeX and word.
Task 3: Creating project abstract. Features to be covered : Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.
Task 4: Creating a Newsletter: Features to be covered: Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

Excel
Excel Orientation: the mentor needs to tell the Importance of MS office or equivalent (FOSS) tool Excel as a Spreadsheet tool. Give the Details of the four tasks and features that would be covered in each Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.
Task 1: Creating a Scheduler: Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text.
Task 2: Calculating GPA: Features to be covered: Cell Referencing, Formulae in excel – average, std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP.
Task 3: Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting.

PowerPoint
Task 1: Students will be working on Basic power point utilities and tools which help them create basic power point presentations.PPT orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in power point.
Task 2: Interactive presentations: Hyperlinks, Inserting – Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 3: Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

REFERENCE BOOKS:

1. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dream tech.
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, WILEY Dream tech.
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. PC Hardware - A Handbook – Kate J. Chase PHI (Microsoft).
5. LaTeX Companion – Leslie Lamport, PHI/Pearson.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – CISCO Press, Pearson Education.

A2510: DISCRETE MATHEMATICS

B.Tech. II Year I Sem.

L T P C
3 0 0 3

Pre-requisite: Mathematical knowledge at pre-university level

Course Objectives: To learn

1. To introduce elementary discrete mathematics for computer science and engineering.
2. To understand formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles, recurrence relations and generating functions.

Course Outcomes: After learning the contents of this paper, the student must be able to

1. Understand and construct precise mathematical proofs.
2. Apply logic and set theory to formulate precise mathematical statements.
3. Analyze and solve counting problems on finite and discrete structures.
4. Describe and manipulate sequences and recurrence relations.
5. Apply graph theory concepts in solving computing problems.

UNIT-I: Mathematical Logic

Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT-II: Set Theory

Set Theory: Introduction, Basic Concepts of Set Theory, Relations and ordering properties. Representation relations, Equivalence relation, POSET, Hasse diagram, Lattice- properties of Lattice. Functions- Types of functions, Inverse of a function.

UNIT-III: Algebraic Structures

Introduction, Algebraic Systems, Semigroups and Monoids and groups.

Recurrence relations: First order recurrence relation, second and higher order linear homogeneous and non-homogeneous recurrence relations.

Generating functions: Method of generating functions for first and second order recurrence relation.

UNIT-IV: Elementary Combinatorics

Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, Principle of Inclusion and Exclusion.

UNIT-V: Graph Theory

Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st Edition.

2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe L. Mott, Abraham Kandel, Theodore P. Baker, Prentice Hall of India, 2nd Edition.

REFERENCE BOOKS:

1. Discrete and Combinatorial Mathematics – An Applied Introduction: Ralph P. Grimaldi, Pearson Education, 5th Edition.

2. Discrete Mathematical Structures: Thomas Koshy, Tata McGraw Hill Publishing Co.

A2511: COMPUTER ORGANIZATION AND ARCHITECTURE

B.Tech. II Year I Sem. **L T P C**
3 0 0 3

Pre-requisites: A Course on IT Workshop

Course Objectives:

1. The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
2. It begins with 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, micro programmed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.
4. To enhance safety awareness and proper handling of workshop equipment.
5. To build a foundational understanding of industrial production and fabrication.

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

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

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 Micro programmed Control

Control memory, Address sequencing, micro program example, design of control unit. Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, 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.

TEXT BOOK:

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

REFERENCE BOOKS:

1. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.
2. Computer Organization and Architecture – William Stallings, Sixth Edition, Pearson/PHI.
3. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.

A2512: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech. II Year I Sem. **L T P C**
3 0 0 3

Pre-requisite: A course on Programming for Problem Solving

Course Objectives:

1. To Understand the basic object-oriented programming concepts and apply them in problem solving.
2. To Illustrate inheritance concepts for reusing the program.
3. To Demonstrate multitasking by using multiple threads and event handling.
4. To Develop data-centric applications using JDBC.
5. To Understand the basics of java console and GUI based programming.

Course Outcomes:

After completion of course, the student will be able to:

1. Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
2. Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.
3. Use multithreading concepts to develop inter process communication.
4. Understand the process of graphical user interface design and implementation using AWT or swings.
5. Develop applets that interact abundantly with the client environment and deploy on the server.

UNIT–I:

Object oriented thinking and Java Basics Need for oop paradigm, summary of oop concepts, coping with complexity, abstraction mechanisms. History of Java, Java buzzwords, data types, variables, scope and lifetime of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring String class.

UNIT–II:

Inheritance, Packages and Interfaces Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super keyword uses, using final keyword with inheritance, polymorphism- method overriding, abstract classes, the Object class. Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

UNIT–III:

Exception handling and Multithreading Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads.

UNIT–IV:

Exploring String class, Object class, Exploring java.io package.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JFrame and JComponent, JLabel, ImageIcon, JTextField, JButton, JCheckBox, JRadioButton, JList, JComboBox, Tabbed Panes, Scroll Panes, Trees, and Tables. Menu Basics, Menu related classes - JMenuBar, JMenu, JMenuItem, JCheckBox MenuItem, JRadioButton MenuItem, JSeperator. creating a popup menu.

UNIT–V:

JDBC: Connecting to Database, JDBC Type 1 to 4 Drivers, connecting to a database, querying a database and processing the results, updating data with JDBC.

Collections & Stream API: Collection API, ArrayList, Set, Map, Comparator vs Comparable, Need of Stream API, Map, Filter, Reduce, Sorted.

TEXT BOOKS:

1. Java the complete reference, 13th edition, Herbert schildt, Dr. Denny Coward, Mc Graw Hill.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John Wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
6. Core Java 2, Vol 2, Advanced Features, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.
7. Object Oriented Programming with Java, R.Buyya, S.T.Selvi, X.Chu, TMH.
8. Java and Object Orientation, an introduction, John Hunt, second edition, Springer.
9. Maurach's Beginning Java2 JDK 5, SPD.

A2514: SOFTWARE ENGINEERING

B.Tech. II Year I Sem. **L T P C**
3 0 0 3

Pre-requisite: Programming for Problem Solving, Data Structures

Course Objectives:

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

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

1. Ability to translate end-user requirements into system and software requirements, using UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

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 modelling, 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.

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill international Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.

REFERENCE BOOKS:

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

A2515: DATABASE MANAGEMENT SYSTEMS

B.Tech. II Year I Sem. **L T P C**
3 0 0 3

Pre-requisites: A course on Data Structures

Course Objectives:

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

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

1. Gain knowledge of fundamentals of DBMS, database design and normal forms.
2. Master the basics of SQL for retrieval and management of data.
3. Be acquainted with the basics of transaction processing and concurrency control.
4. Familiarity with database storage structures and access techniques.

UNIT - I Database System Applications
A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS.
Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model.

UNIT - II Introduction to the Relational Model
Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views. Relational Algebra, Tuple relational Calculus, Domain relational calculus**.**

UNIT - III SQL: QUERIES, CONSTRAINTS, TRIGGERS
form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.
Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, First, Second, Third normal forms, BCNF, lossless join decomposition, multivalued dependencies, Fourth normal form, Fifth normal form.

UNIT - IV Transaction Concept

Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation-Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions.

UNIT - V Data on External Storage

File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM)

B+ Trees: A Dynamic Index Structure.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition. 3rd Edition.
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill.

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education.
3. Introduction to Database Systems, C. J. Date, Pearson Education.
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

A2012: COMPUTATIONAL MATHEMATICS LAB

B.Tech. II Year I Sem. **L T P C**
0 0 2 1

Pre-requisites: Matrices, Iterative methods and ordinary differential equations

Course Objectives:

1. Solve problems of Eigen values and Eigen Vectors using Python/MATLAB.
2. Solution of Algebraic and Transcendental Equations using Python/MATLAB
3. Solve problems of Linear system of equations
4. Solve problems of First-Order ODEs Higher order linear differential equations with constant, coefficients

Course Outcomes:

1. Develop the code to find the Eigen values and Eigen Vectors using Python/MATLAB.
2. Develop the code find solution of Algebraic and Transcendental Equations and Linear system of equations using Python/MATLAB
3. Write the code to solve problems of First-Order ODEs Higher order linear differential equations with constant coefficients

UNIT-I: Eigen values and Eigenvectors
Programs:

- Finding real and complex Eigen values.
- Finding Eigen vectors.

UNIT-II: Solution of Algebraic and Transcendental Equations
Bisection method, Newton Raphson Method
Programs:

- Root of a given equation using Bisection method.
- Root of a given equation Newton Raphson Method.

UNIT-III: Linear system of equations
Jacobi's iteration method and Gauss-Seidal iteration method
Programs:

- Solution of given system of linear equations using Jacobi's method
- Solution of given system of linear equations using Gauss-Seidal method

UNIT-IV: First-Order ODEs

Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling.

Programs:

- Solving exact and non-exact equations
- Solving exponential growth/decay and Newton's law of cooling problems

UNIT-V: Higher order linear differential equations with constant coefficients

Programs:

- Solving homogeneous ODEs
- Solving non-homogeneous ODEs

TEXT BOOKS:

1. MATLAB and its Applications in Engineering, Rajkumar Basal, Ashok Kumar Geo, Manoj Kumar Sharma, Pearson publication.
2. Kenneth A. Lambert, The fundamentals of Python: First Programs, 2011, Cengage Learnings.
3. Think Python First Edition, by Allen B. Downey, Orielly publishing.
4. Introduction to Python Programming, William Mitchell, Povel Solin, Martin Novak et al., NCLab Public Computing, 2012.
5. Introduction to Python Programming, ©Jacob Fredslund, 2007.

REFERENCE BOOKS:

1. An Introduction to Python, John C. Lusth, The University of Alabama, 2011.
2. Introduction to Python, ©Dave Kuhlman, 2008.

A2513: OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

B.Tech. II Year I Sem. **L T P C**
0 0 2 1

Pre-requisite: A course on Programming for Problem Solving

Course Objectives:

1. To write programs using abstract classes.
2. To write programs for solving real world problems using the java collection framework.
3. To write multithreaded programs.
4. To write GUI programs using swing controls in Java.
5. To introduce java compiler and eclipse platform.
6. To impart hands-on experience with java programming.

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

1. Able to write programs for solving real world problems using the java collection framework.
2. Able to write programs using abstract classes.
3. Able to write multithreaded programs.
4. Able to write GUI programs using swing controls in Java.

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint yourself with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
3. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
4. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
5. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
6. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

7. Develop a GUI in Java that displays a simple message.
8. Develop a GUI in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named "Compute" is clicked.
9. Write a java program to display the table using Labels in Grid Layout.
10. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in the selected color. Initially, there is no message shown.
11. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
12. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas.
13. Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after deletion.
14. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
15. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

TEXT BOOKS:

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.

REFERENCE BOOKS:

1. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
2. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

USEFUK LINKS:

1. https://labex.io/tutorials/java-java-methods-and-basic-object-oriented-programming-413809
2. https://www.startertutorials.com/blog/object-oriented-programming-through-java-lab-programs-and-tutorial-videos.html

A2516: DATABASE MANAGEMENT SYSTEMS LAB

B.Tech. II Year I Sem. **L T P C**
0 0 2 1

Pre-requisites: A Course on Data Structures

Course Objectives:
1. Introduce ER data model, database design and normalization
2. Learn SQL basics for data definition and data manipulation

Course Outcomes: At the end of the course the student will be able to
1. Design database schema for a given application and apply normalization
2. Acquire skills in using SQL commands for data definition and data manipulation.
3. Develop solutions for database applications using procedures, cursors and triggers

List of Experiments
1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A. Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)
   B. Nested, Correlated subqueries
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXT BOOKS:
1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill, 3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

REFERENCE BOOKS:
1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition

A2517: DATA VISUALIZATION - TABLEAU

B.Tech. II Year I Sem. **L T P C**
0 0 2 1

Pre-requisite: Basic understanding databases/Excel, and fundamental statistics.

Course Objectives:

1. Effective use of Business Intelligence (BI) technology (Tableau) to apply data visualization
2. To discern patterns and relationships in the data.
3. To build Dashboard applications.
4. To communicate the results clearly and concisely.
5. To be able to work with different formats of data sets.

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

1. Understand How to import data into Tableau.
2. Understand Tableau concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties.
4. Create a Dashboard that links multiple visualizations.
5. Use graphical user interfaces to create Frames for providing solutions to real world problems.

List of Experiments:

1. Understanding Data, what is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization?
2. Getting started with Software using Data file formats, connecting your Data, creating basic charts (line, bar charts, Tree maps), Using the Show me panel.
3. Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.
4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, Formatting specific parts of the view.
5. Editing and Formatting Axes, Manipulating Data and Pivoting.
6. Structuring your data, Sorting and filtering data, Pivoting data.
7. Advanced Visualization Tools: Using Filters, Using the Detail panel, using the Size panels, customizing filters, Using and Customizing tooltips, Formatting your data with colours.
8. Creating Dashboards & Storytelling, creating your first dashboard and Story, Design for different displays, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.
9. File types, publishing Online, Sharing your visualizations, printing, and Exporting.
10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts.

REFERENCE BOOKS:

1. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
2. R Programming for Data Science by Roger D. Peng (References).
3. The Art of R Programming by Norman Matloff Cengage Learning India.

USEFUL LINKS:

1. https://www.udemy.com/course/learn-data-visualization-using-tableau-from-zero-to-hero/?couponCode=PMNVD2025
2. https://studyglance.in/labprograms/powerbiprograms.php
3. https://www.classcentral.com/course/youtube-the-art-of-data-visualization-power-bi-tableau-explained-422585

A2013: ENVIRONMENTAL SCIENCE

B.Tech. II Year I Sem. **L T P C**
1 0 0 1

Pre-requisite: General awareness of environmental issues.

Course Objectives: To learn

1. Understand the components, structure, and functions of ecosystems and their relevance to human society.
2. Comprehend classification, sustainable management, and challenges of natural resources including water, minerals, land, forests, and energy.
3. Grasp the significance, value, and conservation approaches for biodiversity, including threats and legislative frameworks.
4. Analyze types, sources, and impacts of environmental pollution, and learn technological and policy measures for pollution prevention and control.
5. Develop awareness about global environmental challenges, international agreements, and the role of policy, law in sustainable development and EIA.

Course Outcomes: After completion of the course, the student must be able to

1. Understand the structure, function, and significance of ecosystems, including energy flow, biogeochemical cycles, and biodiversity conservation through field experiences.
2. Analyze the classification, utilization, and sustainable management of natural resources, along with alternative energy options.
3. Evaluate biodiversity at genetic, species, and ecosystem levels, its values, threats, and conservation methods under national and international frameworks.
4. Identify types, sources, and impacts of environmental pollution, and apply suitable control technologies while assessing global environmental challenges, protocols and EIA.

UNIT – I: Ecosystems Definition, Scope, and Importance of ecosystem. Classification, structure, and function of an ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycle (nitrogen cycle).

UNIT – II: Natural Resources: Classification of Resources Living and Non-Living resources, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Energy resources: growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT – III: Biodiversity and Biotic Resources Introduction, Definition, genetic, species and ecosystem diversity. India as a mega diversity nation, Hot spots of biodiversity. Forest Act, Wild life Act, Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts. National Biodiversity act.

UNIT – IV: Environmental Pollution and Control Technologies Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. Solid waste: composition and characteristics of e-Waste, plastic waste, Hazardous waste and their management. Global Environmental Issues and Global Efforts: Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

UNIT – V: Environmental Policy and Legislation Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules. EIA – definition, scope and applications. Environmental Management Plan (EMP). Sustainable development goals (SDGs); Global environmental challenges.

TEXT BOOKS:

1. Introduction to Environmental Science by Y. Anjaneyulu, BS. Publications.
2. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
3. Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Text book of Environmental Science and Technology – Dr. M. Anji Reddy 2007, BS Publications.

A2014: COMPUTER ORIENTED STATISTICAL METHODS

B.Tech. II Year II Sem. **L T P C**
3 0 0 3

Pre-requisite: Mathematics courses of first year of study

Course Objectives: To learn

1. The theory of Random variable Probability distributions of single random variables
2. The sampling theory, testing of hypothesis and making statistical inferences
3. The curve fitting, correlation and regression for the given data.

Course Outcomes:

1. Apply the concepts of Random variable and distributions to case studies.
2. Formulate and solve problems involving random variables and apply statistical methods for analyzing experimental data.
3. Apply concept of estimation and testing of hypothesis to case studies.
4. Fit the curve, correlation and regression for the given data.

UNIT-I: Random Variables and Probability Distributions

Concept of a Random Variable – Discrete Probability Distributions – Continuous Probability Distributions – Mean of a Random Variable – Variance of a Random Variable

UNIT-II: Continuous Distributions and sampling

Normal Distribution – Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial Distributions. Fundamental Sampling Distributions: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means, Central Limit Theorem.

UNIT-III: Estimation

Introduction, Statistical Inference, Classical Methods of Estimation – Single Sample: Estimating the mean – Standard error of a point estimate. Two samples: Estimating the difference between two means– Single sample: Estimating a proportion – Two samples: Estimating the difference between two proportions– Two samples: Estimating the ratio of two variances.

UNIT-IV: Tests of Hypotheses (Large and Small Samples)

Statistical Hypotheses: General Concepts – Testing a Statistical Hypothesis. Single sample: Tests concerning a single mean. Two samples: Tests on two mean (Unknown for equal variance). One sample: Test on a single proportion. Two samples: Tests on two proportions. Small samples: t-test for single and two means, Two- sample tests concerning variances: F-test

UNIT-V: Applied Statistics

Curve fitting by the method of least squares, Fitting of straight lines, Second degree parabolas and more general curves, Correlation and Regression, Rank correlation.

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press
2. Miller and Freund's, Probability and Statistics for Engineers, 8th Edition, Pearson Educations.
3. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons, 2004.

A2518: OPERATING SYSTEMS

B.Tech. II Year II Sem.

L	T	P	C
3	0	0	3

Pre-requisite:

1. A course on Computer Programming and Data Structures
2. A course on Computer Organization and Architecture

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, interprocess communication and I/O in Unix

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

1. Control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computers and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT-I:

Operating Systems Introduction

Structures - Simple Batch, Multi programmed, 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.

UNIT-II:

CPU Scheduling

Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec.

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

UNIT-III:

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

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

UNIT-IV:

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

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition– 2005, Pearson Education/PHI.
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI.
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education.
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

A2506: PYTHON PROGRAMMING

B.Tech. II Year II Sem. **L T P C**
3 0 0 3

Pre-requisite: Basic Knowledge of Computers and Programming Logic

Course Objectives:

1. Understand the fundamentals and syntax of Python programming.
2. Develop applications using control structures, functions, and modules.
3. Learn data structures such as lists, tuples, dictionaries, and sets.
4. Gain practical experience with file handling, exceptions, and object-oriented concepts.
5. Develop and debug Python programs for real-time applications.

Course Outcomes:

1. Understand basic syntax, keywords, and control structures in Python.
2. Design Python programs using functions, recursion, and modules.
3. Implement data structures such as lists, tuples, sets, and dictionaries.
4. Perform file operations and handle exceptions effectively.
5. Apply object-oriented programming principles and develop modular code.
6. Solve complex real-time problems using Python programming skills.

UNIT – I: Python Basics

Introduction to Python, features, applications, Python interpreter and IDEs, Variables, data types, and operators, Input/output functions, Control structures: if, if-else, nested if, loops (for, while), break and continue.

UNIT – II: Functions and Modules

Defining and calling functions, Function arguments and return values, Recursive functions, Lambda functions, Modules and packages, built-in and user-defined modules, Introduction to Python Standard Library**.**

UNIT – III: Data Structures in Python

Lists: operations, list comprehensions, Tuples: accessing, slicing, and unpacking, Sets and its operations, Dictionaries: creation, operations, and methods, Iterating through data structures.

UNIT – IV: File Handling, Exception Handling, and Regular Expressions

Reading and writing text/binary files, File methods and context manager, Handling exceptions using try, except, finally, Raising exceptions, Regular expressions: match, search, findall, sub.

UNIT – V: Object-Oriented Programming (OOP) in Python & Extended Modules Classes and objects, Constructors and destructors, Inheritance and polymorphism, Encapsulation and data hiding, Method overloading and overriding, Introduction to decorators and iterators. Numpy, pandas, matplotlib, seaborn, Scikit-learn.

TEXT BOOKS:

1. Reema Thareja, “Python Programming: Using Problem Solving Approach,” Oxford University Press.
2. Dr. R. Nageswara Rao, “Core Python Programming,” Dreamtech Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist,” O’Reilly.

REFERENCE BOOKS:

1. Mark Lutz, “Learning Python,” O’Reilly Media.
2. Wesley J. Chun, “Core Python Applications Programming,” Pearson.
3. Charles Dierbach, “Introduction to Computer Science Using Python,” Wiley.
4. Zed A. Shaw, “Learn Python the Hard Way,” Addison-Wesley.

USEFUL LINKS:

1. https://www.w3schools.com/python
2. https://onlinecourses.nptel.ac.in/noc22_cs26/preview
3. https://www.javatpoint.com/python-tutorial
4. https://www.tutorialspoint.com/python
5. https://www.programiz.com/python-programming
6. https://www.geeksforgeeks.org/python-programming-language/

A2520: COMPUTER NETWORKS

B.Tech. II Year II Sem. **L T P C**
3 0 0 3

Pre-requisite: A course on Programming for problem solving, A course on Data Structures

Course Objectives: To learn

1. Equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

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

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the ISO-OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT – I: Introduction

Introduction: The Internet, Protocol, Network Edge, Access Networks, Network Core, Packet Switching, Circuit Switching, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol reference models: ISO-OSI, TCP/IP, Types of Network attacks, History of Computer Networking and the Internet.

UNIT – II: Application Layer

Application Layer: Principles of Network Applications, Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, SMTP, DNS, Peer-to-Peer Applications, Socket Programming: Creating Network Applications.

UNIT – III: Transport Layer

Transport Layer: Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N (GBN),

Selective Repeat (SR), Connection-Oriented Transport: TCP, The TCP Connection, Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control, TCP Congestion Control, Fairness.

UNIT – IV: Network Layer: Data and Control plane

Network Layer: Data and Control plane, Forwarding and Routing 308, Network Service Models, Virtual Circuit and Datagram Networks, Router working, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing, Internet Control Message Protocol (ICMP), IPv6, IP Security, Routing Algorithms- The Link-State (LS) Routing Algorithm, The Distance- Vector (DV) Routing Algorithm, Hierarchical Routing, Routing in the Internet-Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing: BGP, Broadcast and Multicast Routing, Broadcast Routing Algorithms, Multicasting.

UNIT – V: The Link Layer

The Link Layer: The Services Provided by the Link Layer, Error-Detection and -Correction Techniques- Parity Checks, Checksum Methods, Cyclic Redundancy Check (CRC), Hamming code, Multiple Access Links and Protocols, Channel Partitioning Protocols, Random Access Protocols, Taking-Turns Protocols, DOCSIS: The Link-Layer Protocol for Cable Internet Access, Switched Local Area Networks, Link-Layer Addressing and ARP, Ethernet, Link-Layer Switches, Virtual Local Area Networks (VLANs), Link Virtualization-Multiprotocol Label Switching (MPLS), Data Center Networking, A Day in the Life of a Web Page Request. Wireless network characteristics, Wireless LAN.

TEXT BOOKS:

1. Computer Networking: A Top-Down Approach – James F. Kurose, Keith W. Ross, Pearson.
2. Computer Networks – Andrew S Tanenbaum, David J. Wetherall, 5th Edition. Pearson/PHI.

REFERENCE BOOKS:

1. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

USEFUL LINKS:

1. https://www.geeksforgeeks.org/computer-networks/computer-network-tutorials/
2. https://www.w3schools.blog/computer-network
3. https://www.tpointtech.com/computer-network-tutorial
4. https://www.tutorialspoint.com/computer_fundamentals/computer_networking.htm

A2522: WEB TECHNOLOGIES

B.Tech. II Year II Sem.

L	T	P	C
3	0	0	3

Pre-requisite: A course on Object Oriented Programming through JAVA

Course Objectives:

1. To understand static website design using HTML.
2. To understand website design using PHP for server-side scripting.
3. To design websites using JavaScript, PHP, and servlets.
4. To design dynamic business websites using JSP.
5. To understand and validate client-side forms.

Course Outcomes: After completion of this course, the student will be able to

1. Design static and interactive web pages for a given application.
2. Demonstrate understanding of server-side scripting with PHP.
3. Analyze the working of Java servlets.
4. Create the database and perform operations using JDBC.
5. Demonstrate server-side scripting and database connectivity using JSP.
6. Validate inputs using JavaScript.

UNIT-I: Introduction to Markup Languages

HTML Common tags: List, Tables, images, forms, Frames; Cascading Style sheets;

XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

UNIT-II: Server-side Scripting with PHP

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies.

File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT-III: Server-side Programming with Servlets

Introduction to Servlets: Common Gateway Interface (CGI), Life cycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

UNIT-IV: Server-side Programming with 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.

UNIT-V: Client-side Scripting with Javascript

Client-side Scripting: Introduction to Javascript, Javascript language – declaring variables, scope of variables, functions. event handlers (onclick, onsubmit etc.), Document Object Model, Form validation.

TEXT BOOKS:

1. Uttam K Roy, Web Technologies, Oxford University Press.
2. Steven Holzner, The Complete Reference PHP, Tata McGraw-Hill.

REFERENCE BOOKS:

1. Chris Bates, Web Programming, building internet applications, 2nd edition, Wiley Dreamtech.
2. Hans Bergsten, Java Server Pages, SPD O'Reilly.
3. D.Flanagan, Java Script.
4. Jon Duckett, Beginning Web Programming, WROX.
5. R.W.Sebesta, Programming world wide web, Fourth Edition, Pearson.
6. Dietel and Nieto, Internet and World Wide Web — How to program.

USEFUL LINKS:

1. https://www.tutorialspoint.com/php/index.htm
2. https://www.w3schools.com/

A2015: INNOVATION AND ENTREPRENEURSHIP

B.Tech. II Year II Sem. **L T P C**
2 0 0 2

Course Objectives:

1. To familiarize on the basic concepts of innovation, entrepreneurship and its importance.
2. To Identify and analyze the process of problem-opportunity identification, market segmentation, and idea generation techniques.
3. To initiate prototype development and understand minimum viable product.
4. To develop initial Business and financial planning and Go-to-Market strategies.
5. To impart knowledge on establishing startups, venture pitching and IPR.

Course Outcomes:

1. Understand the entrepreneurship and the entrepreneurial process and its significance in economic development.
2. Assess the problem from an industry perspective and generate solutions using the design thinking principles.
3. Assess market competition, estimate market size, and develop a prototype.
4. Analyze Business and financial planning models and Go-to-Market strategies.
5. Able to build a start-up, register IP and identify funding opportunities.

UNIT-I: Fundamentals of Innovation and Entrepreneurship

Innovation: Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation.

Entrepreneurship: Introduction, types of entrepreneurship attributes, mindset of entrepreneurial and intrapreneurial leadership, Role of entrepreneurs in economic development. Woman Entrepreneurship, Importance of on-campus startups. Understanding to build entrepreneurial mindset, attributes and networks individuals while on campus. Core Teaching Tool: Simulation, Game, Industry Case Studies (Personalized for students – 16 industries to choose from), Venture Activity.

UNIT-II: Problem and Customer Identification

Identification of gap, problem, analyzing the problem from a industry perspective, real-world problems, market and customer segmentation, validation of customer problem fit, Iterating problem-customer fit, Competition and Industry trends mapping and assessing initial opportunity, Porter's Five Force Model. Idea generation, Ideation techniques: Brainstorming, Brain writing, Round robin, and SCAMPER, Design thinking principles, Mapping of solution to problem. Core Teaching Tool: Several types of activities including: Class, game, Gen AI, 'Get out of the Building' and Venture Activity.

UNIT-III: Opportunity assessment and Prototype development

Identify and map global competitors, review industry trends, and understand market sizing: TAM, SAM, and SOM. Assessing scope and potential scale for the opportunity. Understanding prototyping and Minimum Viable Product (MVP). Developing a prototype: Testing, and validation. Core Teaching Tool: Venture Activity, no-code Innovation tools, Class activity.

UNIT-IV: Business & Financial Models

Introduction to Business Model and types, Lean Canvas Approach: 9-block lean canvas model, building lean canvas for your startup. Business planning: components of Business plan- Sales plan, People plan and financial plan. Financial Planning: Types of costs, preparing a financial plan for profitability using a financial template, understanding the basics of Unit economics, Economies of Scale and analyzing financial performance. Go-To-Market (GTM) approach – Selecting the Right Channel, creating digital presence, and building customer acquisition strategy. Core Teaching Tool: Founder Case Studies – Sama and Securely Share; Class activity and discussions; Venture Activities.

UNIT-V: Startups and IPR

Startup requirements, building founding team members and mentors, pitch preparation, start-up registration process, funding opportunities and schemes, institutional support to entrepreneurs, startup lifecycle, documentation, legal aspects in startup, venture pitching readiness, National Innovation Startup Policy (NISP) and its features. Patents, Designs, Patentability, Procedure for grants of patents. Indian Scenario of Patenting, International Scenario: International cooperation on Intellectual Property. Patent Rights: Scope of Patent Rights. Copyright, trademark, and GI. Licensing and transfer of technology. Core Teaching Tool: Expert talks; Cases; Class activity and discussions; Venture Activities.

SUGGESTED READINGS:

1. John R Bessant, Joe Tidd, Innovation and Entrepreneurship, 4E, Wiley, Latest Edition.
2. Ajay Batra, The Stratup Launch Book- A Practical Guide for Launching Customer Centric Ventures, Wiley, 2020.
3. Entrepreneurship Development and Small Business Enterprises, Poornima M Charantimath, 3E, Pearson, 2018.
4. D.F. Kuratko and T.V. Rao, Entrepreneurship: A South-Asian Perspective, Cengage Learning, 2013.
5. Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, Sabyasachi Sinha (2020). Entrepreneurship, McGrawHill, 11th Edition.
6. NISP -Brochure inside pages - startup_policy_2019.pdf.

A2519: OPERATING SYSTEMS LAB

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Prerequisites: A course on Programming for Problem Solving, Computer Organization and Architecture

Course Objectives:

1. To provide an understanding of the design aspects of operating system concepts through simulation.
2. Introduce basic Unix commands, system call interface for process management, inter process communication and I/O in Unix.

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

1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. Able to implement C programs using Unix system calls.

List of Experiments

1. Write C programs to simulate the following CPU Scheduling algorithms a) FCFS b) SJF c) Round Robin d) priority.
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir).
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms a) Pipes b) FIFOs c) Message Queues d) Shared Memory.
6. Write C programs to simulate the following memory management techniques a) Paging b) Segmentation.
7. Write C programs to simulate Page replacement policies a) FCFS b) LRU c) Optimal.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the Unix environment, W.R.Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition– 2005, Pearson Education/PHI.
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.

A2521: COMPUTER NETWORKS LAB

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Prerequisites: A course on Programming for Problem Solving, Object Oriented Programming through JAVA

Course Objectives:

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance.
3. To analyze the traffic flow and the contents of protocol frames.

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

1. Implement data link layer farming methods.
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer.
5. To be able to work with different network tools.

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijkstra's algorithm to compute the shortest path through a network.
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption.
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. Wireshark, Nmap
    - Packet Capture Using Wireshark.
    - Starting Wireshark.
    - Viewing Captured Traffic.
    - Analysis and Statistics & Filters.
    - How to run Nmap scan.
    - Operating System Detection using Nmap.

11. Do the following using NS2 Simulator
    - NS2 Simulator-Introduction.
    - Simulate to Find the Number of Packets Dropped.
    - Simulate to Find the Number of Packets Dropped by TCP/UDP.
    - Simulate to Find the Number of Packets Dropped due to Congestion.
    - Simulate to Compare Data Rate & Throughput.
    - Simulate to Plot Congestion for Different Source/Destination.
    - Simulate to Determine the Performance with respect to Transmission of Packets.

TEXT BOOKS:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI.
2. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

A2507: PYTHON PROGRAMMING LAB

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Pre-requisite: Basic Knowledge of Computers and Programming Logic

Course Objectives:

1. Understand the fundamentals and syntax of Python programming.
2. Develop applications using control structures, functions, and modules.
3. Learn data structures such as lists, tuples, dictionaries, and sets.
4. Gain practical experience with file handling, exceptions, and object-oriented concepts.
5. Develop and debug Python programs for real-time applications.

Course Outcomes:

1. Understand basic syntax, keywords, and control structures in Python.
2. Design Python programs using functions, recursion, and modules.
3. Implement data structures such as lists, tuples, sets, and dictionaries.
4. Perform file operations and handle exceptions effectively.
5. Apply object-oriented programming principles and develop modular code.
6. Solve complex real-time problems using Python programming skills.

Week-1

1. Calculate the total and average of marks scored in five subjects.
2. Determine if a year is a leap year.
3. Accept a string and count vowels, consonants, digits, and special characters.
4. Accept a number and determine if it is prime.
5. Create a calculator using functions.

6. Simulate ATM transactions (withdrawal, balance check).
7. Generate Fibonacci series up to N terms using recursion.
8. Convert temperature from Celsius to Fahrenheit and vice versa.

Week-2

1. Accept a string and reverse it using a function.
2. Display multiplication tables from 1 to N using nested loops.
3. Accept a string and count the frequency of each character.
4. Perform matrix addition and multiplication.
5. Find the factorial of a number using recursion.

Week-3

1. Store and display student details using dictionaries.
2. Validate a user-defined password policy.
3. Accept a sentence and count the number of words.
4. Create a login authentication system.
5. Create a contact book using dictionary and functions.
6. Write a program to simulate a library management system.
7. Accept a list of numbers and remove duplicates.

Week-4

1. Implement a simple voting system.
2. Accept a list and sort it in ascending and descending order.
3. Accept N names and display them in alphabetical order.
4. Accept N email addresses and validate them using regex.
5. Implement a pattern printing program.
6. Read a text file and count the number of lines and words.
7. Convert binary number to decimal and vice versa.
8. Create a mini-billing system.

Week-5

1. Accept a list of tuples and sort them based on the second element.
2. Implement a marks analysis system for N students.
3. Develop a simple calculator using lambda expressions.
4. Create a quiz application using dictionaries.
5. Create a dictionary with product details and implement search.
6. Accept date input and validate it using regex.
7. Implement a string compression and decompression tool.
8. Accept file name and display contents in reverse order.
9. Track attendance of students using sets.

Week-6

1. Accept two matrices and perform transpose.
2. Merge two sorted lists into a single sorted list.
3. Develop a mini hospital appointment booking system.
4. Accept a file and replace all spaces with underscores.
5. Simulate a digital clock.
6. Create a bank account class with deposit, withdraw, and balance.
7. Accept a number and check if it is an Armstrong number.
8. Create a basic weather report display using dictionaries.
9. Create a bus ticket booking application.
10. Simulate a parking system using OOP.

Week-7

1. Develop a mini email sender simulator (mock only).
2. Manage employee records using file handling.
3. Find common elements in two lists using set operations.
4. Create a menu-driven grocery store billing application.
5. Convert a paragraph to title case.
6. Accept a mobile number and validate its format using regex.

7. Create a dictionary of employees and find the one with max salary.
8. Implement stack and queue using lists.
9. Read and write CSV files.
10. Accept coordinates and calculate distance between two points.
11. Simulate a random OTP generator.
12. Create a grade calculator with dynamic marks entry.

TEXT BOOKS:

1. Reema Thareja, “Python Programming: Using Problem Solving Approach,” Oxford University Press.
2. Dr. R. Nageswara Rao, “Core Python Programming,” Dreamtech Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist,” O’Reilly.

REFERENCE BOOKS:

1. Mark Lutz, “Learning Python,” O’Reilly Media.
2. Wesley J. Chun, “Core Python Applications Programming,” Pearson.
3. Charles Dierbach, “Introduction to Computer Science Using Python,” Wiley.
4. Zed A. Shaw, “Learn Python the Hard Way,” Addison-Wesley.

USEFUL LINKS:

1. https://www.w3schools.com/python
2. https://onlinecourses.nptel.ac.in/noc22_cs26/preview
3. https://www.javatpoint.com/python-tutorial
4. https://www.tutorialspoint.com/python
5. https://www.programiz.com/python-programming
6. https://www.geeksforgeeks.org/python-programming-language/

A2523: WEB TECHNOLOGIES LAB

B.Tech. II Year II Sem. **L T P C**
0 0 2 1

Pre-requisite: Object Oriented Programming through JAVA

Course Objectives:

1. Developing static and dynamic web pages for an online bookstore.
2. Implementing server-side logic using PHP, Servlets, and JSP.
3. Managing data using XML and database systems.
4. Validating user input through client-side scripting.

Course Outcomes: After completion of this course, the student will be able to

1. Design a multi-page static website resembling professional e-commerce platforms.
2. Write PHP scripts for string manipulation, array processing, and file I/O operations.
3. Deploy and configure TOMCAT web servers for servlet-based applications.
4. Implement session management using cookies and shopping cart logic.
5. Apply MVC architecture to develop database-driven web applications.
6. Perform client-side validation for registration and payment forms.

Markup Languages

1.Develop static pages of an online book store resembling www.amazon.com, including: Home page, Registration/Login, User Profile, Books catalog, Shopping Cart, Payment by credit card, and Order Confirmation.

Server-side Scripting with PHP

2. Write a PHP script to print prime numbers between 1-50.
3. PHP scripts for string operations: length, word count, reversing, and searching.
4. PHP script to merge two arrays and sort them as numbers in descending order.
5. PHP script to read data from one file and write it into another.

Server-side Programming with Servlets

6. Install TOMCAT web server and convert static pages into dynamic pages using servlets and cookies (storing user info in web.xml).
7. Create/save an XML document with 10 users; write a program to retrieve details by User Id.

Server-side Programming with JSP

8. Redo the bookstore task using JSP and MVC architecture; create a database for user and book information to load the catalog dynamically.

Client-side Scripting with JavaScript

9. Validate Registration, Login, User Profile, and Payment pages using JavaScript.

TEXT BOOK:

1. Jeffrey C. Jackson, *WEB TECHNOLOGIES: A Computer Science Perspective*, Pearson Education.

REFERENCE BOOKS:

1. S. Keshav, *An Engineering Approach to Computer Networks*, 2nd Edition, Pearson Education.
2. Deitel H.M. and Deitel P.J., *Internet and World Wide Web How to program*, 4th Edition, 2012.
3. James Keogh, *J2EE: The complete Reference*, McGraw-Hill.
4. Bai and Ekedhi, *The Web Warrior Guide to Web Programming*, Thomson.
5. Paul Dietel and Harvey Deitel, *Java How to Program*, 8th Edition, Prentice Hall of India.
6. *Web technologies, Black Book*, Dreamtech press.
7. Gopalan N.P. and Akilandeswari J., *Web Technology*, Prentice Hall of India.

A2524: UI DESIGN- FLUTTER

B.Tech. II Year II Sem.

L	T	P	C
0	0	2	1

Pre-requisite: Basic programming knowledge (preferably in Dart/Java/Kotlin), understanding of OOP concepts.

Course Objectives:

1. Learns to Implement Flutter Widgets and Layouts.
2. Understands Responsive UI Design and with Navigation in Flutter.
3. Knowledge on Widgets and customize widgets for specific UI elements, Themes.
4. Understand to include animation apart from fetching data.

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

1. Implements Flutter Widgets and Layouts.
2. Responsive UI Design and with Navigation in Flutter.
3. Create custom widgets for specific UI elements and also Apply styling using themes and custom styles.
4. Design a form with various input fields, along with validation and error handling.
5. Fetches data and write code for unit Test for UI components and also animation.

List of Experiments:

1. a) Install Flutter and Dart SDK.
   b) Write a simple Dart program to understand the language basics.
2. a) Explore various Flutter widgets (Text, Image, Container, etc.).
   b) Implement different layout structures using Row, Column, and Stack.
3. a) Design a responsive UI that adapts to different screen sizes.
   b) Implement media queries and breakpoints for responsiveness.
4. a) Set up navigation between different screens using Navigator.
   b) Implement navigation with named routes.
5. a) Learn about stateful and stateless widgets.
   b) Implement state management using setState and Provider.
6. a) Create custom widgets for specific UI elements.
   b) Apply styling using themes and custom styles.
7. a) Design a form with various input fields.
   b) Implement form validation and error handling.
8. a) Add animations to UI elements using Flutter's animation framework.
   b) Experiment with different types of animations (fade, slide, etc.).
9. a) Fetch data from a REST API.
   b) Display the fetched data in a meaningful way in the UI.
10. a) Write unit tests for UI components.
    b) Use Flutter's debugging tools to identify and fix issues.

TEXT BOOKS:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development, 1st edition, Wrox publisher.

USEFUL LINKS

1. https://github.com/srinu2003/Flutter-Lab
2. https://www.udemy.com/course/flutter-ui-coding-course-for-designers-and-beginners/?couponCode=PMNVD2025