COURSE STRUCTURE

(I to IV years)

And

DETAILED SYLLABI

(I to II Years)

of

R25 Regulation

For

B.Tech. Computer Science and Engineering

(Applicable for the Batch admitted from 2025-26)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

(An Autonomous Institution approved by UGC and affiliated to JNTUH)

Yamnampet, Ghatkesar, Hyderabad - 501 301

August 2025

Contents

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY 4

B.Tech. in COMPUTER SCIENCE AND ENGINEERING 4

Course Structure & Syllabus of I, II Years (A25 Regulations) 4

I Year I Semester 4

I Year II Semester 4

II YEAR I SEMESTER 5

II YEAR II SEMESTER 5

Syllabus of I Year I Sem 6

MATRICES AND CALCULUS 6

ADVANCED ENGINEERING PHYSICS 8

ENGLISH FOR SKILL ENHANCEMENT 11

ELECTRONIC DEVICES AND CIRCUITS 14

PROGRAMMING FOR PROBLEM SOLVING 16

ENVIRONMENTAL SCIENCE 18

ADVANCED ENGINEERING PHYSICS LAB 20

ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB 21

PROGRAMMING FOR PROBLEM SOLVING LAB 24

ENGINEERING WORKSHOP 28

Syllabus of I Year II Sem 30

ORDINARY DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS 30

ENGINEERING CHEMISTRY 32

ENGINEERING DRAWING AND COMPUTER AIDED DRAFTING 34

BASIC ELECTRICAL ENGINEERING 36

DATA STRUCTURES 38

ENGINEERING CHEMISTRY LAB 40

BASIC ELECTRICAL ENGINEERING LAB 42

DATA STRUCTURES LAB 44

PYTHON PROGRAMMING LAB 46

IT WORKSHOP 49

Syllabus of II Year I Sem 51

DISCRETE MATHEMATICS 51

COMPUTER ORGANIZATION AND ARCHITECTURE 53

OBJECT ORIENTED PROGRAMMING THROUGH JAVA 55

SOFTWARE ENGINEERING 57

DATABASE MANAGEMENT SYSTEMS 59

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB 61

SOFTWARE ENGINEERING LAB 64
DATABASE MANAGEMENT SYSTEMS LAB 66
UNIVERSAL HUMAN VALUES 67
CODING SKILLS 69
Innovation and Entrepreneurship-I (Engineering Exploration) 74
Syllabus of II Year II Sem 75
COMPUTER ORIENTED STATISTICAL METHODS 75
COMPUTER NETWORKS 77
OPERATING SYSTEMS 79
DESIGN AND ANALYSIS OF ALGORITHMS 81
Business Economics and Financial Analysis 83
COMPUTATIONAL MATHEMATICS LAB 85
COMPUTER NETWORKS LAB 87
OPERATING SYSTEMS LAB 89
REACT JS/ NODE JS/ EXPRESS JS/ MONGO DB 90
Innovation and Entrepreneurship-II (Prototype Realization) 92

SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY

B.Tech. in COMPUTER SCIENCE AND ENGINEERING

Course Structure & Syllabus of I, II Years (A25 Regulations)

Applicable from AY 2025-26 Batch

I Year I Semester

S. No	Course Category	Course Code	Course Title	L	T	P	Credits	CIE	SEE
1.	BSC	25HC07	Matrices and Calculus	3	1	0	4	40	60
2.	BSC	25HC03	Advanced Engineering Physics	3	0	0	3	40	60
3.	HSC	25HC01	English for Skill Enhancement	3	0	0	3	40	60
4.	ESC	25CC01	Electronic Devices and Circuits	3	0	0	3	40	60
5.	PCC/CSC	25FC01	Programming for Problem Solving	3	0	0	3	40	60
6.	MC	25HC05	Environmental Science	1	0	0	1	40	60
7.	BSC	25FC01	Advanced Engineering Physics Lab	0	0	2	1	40	60
8.	HSC	25HC61	English Language Communication Skills Lab	0	0	2	1	40	60
9.	PCC/CSC	25FC61	Programming for Problem Solving Lab	0	0	2	1	40	60
10.	MEC	25BC61	Engineering Workshop	0	0	2	1	40	60
			Total Credits	**16**	**1**	**8**	**21**	**400**	**600**

I Year II Semester

S. No.	Course Category	Course Code	Course Title	L	T	P	Credits	CIE	SEE
1.	BSC	25HC08	Ordinary Differential Equations and Vector Calculus	3	0	0	3	40	60
2.	BSC	25HC04	Engineering Chemistry	3	0	0	3	40	60
3.	ESC/MEC	25BC01	Engineering Drawing and Computer Aided Drafting	2	0	2	3	40	60
4.	ESC	25AC47	Basic Electrical Engineering	3	0	0	3	40	60
5.	PCC/ESC	25EC01	Data Structures	3	0	0	3	40	60
6.	BSC	25HC65	Engineering Chemistry Lab	0	0	2	1	40	60
7.	ESC	25AC77	Basic Electrical Engineering Lab	0	0	2	1	40	60
8.	PCC/ESC	25EC61	Data Structures Lab	0	0	2	1	40	60
9.	PCC/CSCs	25FC62	Python Programming Lab	0	0	2	1	40	60
10.	HSC/CSC	25FC63	IT Workshop	0	0	2	1	40	60
			Total Credits	**14**	**0**	**12**	**20**	**400**	**600**

II YEAR I SEMESTER

S. No.	Course Category	Course Code	Course Title	L	T	P	Credits	CIE	SEE
1.	PCC	25FC02	Discrete Mathematics	3	0	0	3	40	60
2.	PCC	25CC47	Computer Organization and Architecture	3	0	0	3	40	60
3.	PCC	25EC02	Object Oriented Programming through java	3	0	0	3	40	60
4.	PCC	25EC03	Software Engineering	3	0	0	3	40	60
5.	PCC	25FC03	Database Management Systems	3	0	0	3	40	60
6.	PCC	25EC62	Object Oriented Programming through java Lab	0	0	2	1	40	60
7.	PCC	25EC63	Software Engineering Lab	0	0	2	1	40	60
8.	PCC	25FC64	Database Management Systems Lab	0	0	2	1	40	60
9.	MC	25HC06	Universal Human Values	1	0	0	1	40	60
10.	SDC	25EC04	Coding Skills	0	0	2	1	40	60
11.	PS	25CC60	Innovation and Entrepreneurship-I (Engineering Exploration)	0	0	2	1	40	60
			Total Credits	**16**	**0**	**10**	**21**	**440**	**660**

II YEAR II SEMESTER

S. No.	Course Category	Course Code	Course Title	L	T	P	Credits	CIE	SEE
1.	BSC	25HC13	Computer oriented Statistical Methods	3	0	0	3	40	60
2.	PCC	25EC05	Computer Networks	3	0	0	3	40	60
3.	PCC	25EC06	Operating Systems	3	0	0	3	40	60
4.	PCC	25EC07	Design and Analysis of Algorithms	3	0	0	3	40	60
5.	HSC	25ZC01	Business Economics and Financial Analysis	3	0	0	3	40	60
6.	BSC	25HC66	Computational Mathematics Lab	0	0	2	1	40	60
7.	PCC	25EC64	Computer Networks Lab	0	0	2	1	40	60
8.	PCC	25EC65	Operating Systems Lab	0	0	2	1	40	60
9.	SDC	25FC10	React JS, Node JS, Express JS, Mongo DB	0	0	2	1	40	60
10.	PS	25CC61	Innovation and Entrepreneurship-II (Prototype Realization)	0	0	2	1	40	60
			Total Credits	**15**	**0**	**10**	**20**	**400**	**600**

Syllabus of I Year I Sem

MATRICES AND CALCULUS

(Common to all Branches)

L	T	P	C
3	1	0	4

Code: 25HC07
B. Tech. I Year I Semester
Pre Requisites: Mathematics Knowledge at Pre-University Level

Course Objectives:

1. The concepts of linear algebra
2. Differential calculus
3. The concepts of multivariable calculus

Course Outcomes:

1. Solving the system of linear equations by checking the consistency and evaluation of eigen values and eigen vectors.
2. Diagonalize the matrix and apply Cayley Hamilton Theorem to obtain the higher powers of a matrix and also identify the nature of the quadratic form.
3. Apply appropriate mean value theorems to obtain the mean values and find the power series expansion of a function using Taylor's series.
4. Compute the Jacobian transformation, the extreme values of a multivariable function.
5. Evaluate double integrals using change of order of integration and change of variables, evaluation of triple integrals.

UNIT-I: Matrices-I: Rank of a matrix, Echelon form, Normal form, Inverse of Non-singular matrices by Gauss-Jordan method. System of linear equations; Consistency of Homogeneous and Non-Homogeneous equations, Gauss Seidal Iteration Method. Evaluation of Eigen values and Eigen vectors for a given matrix. [10 hours]

UNIT-II: Matrices-II: Diagonalization of matrices; Cayley-Hamilton Theorem (without proof) and its applications in finding higher powers & inverse of a matrix; Quadratic forms-Nature, Rank, Index and Signature of a Quadratic form. Reduction of Quadratic form to canonical form by Orthogonal Transformation. [10 hours]

UNIT-III: Single Variable Calculus: Rolle's theorem and Lagrange's Mean value theorem with their Geometrical Interpretation, Cauchy's Mean value theorem (without proof); Taylor's and Maclaurin's series (without proof) and their application for series expansions of standard functions. [Hours: 6 hours]

Curve Tracing: Curve tracing in Cartesian coordinates [Hours: 4 hours]

UNIT-IV: Functions of several variables: Definitions of Limit and continuity. Partial Differentiation; Euler's Theorem; Total derivative, Chain Rule, Jacobian transformation, Functional dependence & independence, Maxima and Minima of two variables functions (with and without constraints). [Hours: 10 hours]

UNIT-V: Multiple Integrals with Applications: Evaluation of Double Integrals (Cartesian and polar coordinates); Change of order of integration (only Cartesian form), Change of variables (Cartesian to polar). Evaluation of Triple integrals (Cartesian coordinates). Applications: Areas (by double integrals) and volumes (by triple integrals).[10 hours]

TEXT BOOKS:

1. R K Jain and S R K Iyengar Advanced Engineering Mathematics, Narosa Publications.
2. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
3. Alan Jeffery, Advanced Engineering Mathematics, Academic Press
4. B. V. Ramana, Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
5. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

ADVANCED ENGINEERING PHYSICS

Course Code: 25HC03
B.Tech I Year I Sem
Prerequisites: 10 + 2 Physics

L	T	P	C
3	0	0	3

Course Objectives:

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

Course Outcomes:

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

UNIT-I: Crystallography & Materials 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 Scherrer’s formula, scanning electron microscopy (SEM): block diagram, working principle.

UNIT-II: Laser and Fibre Optics

Introduction to laser, characteristics of laser, Einstein coefficients and their relations, metastable state, population inversion, pumping, lasing action, Ruby laser, He-Ne laser, CO_2 laser, semiconductor diode laser, applications: Bar code scanner, LIDAR for autonomous 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.

UNIT-III: 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): electronics, ionic & orientation; ferroelectric, piezoelectric, pyroelectric materials and their applications: Ferroelectric Random-Access Memory (Fe-RAM), load cell and fire sensor.

UNIT-IV: 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 bandgap, classification of solids, concept of discrete energy levels and quantum confinement in nanomaterials.

UNIT-V: 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, evolution of quantum systems, quantum measurements, entanglement, quantum gates, challenges and advantages of quantum computing over classical computation, quantum algorithms: Deutsch-Jozsa, Shor, Grover.

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

REFERENCES:

1. JozefGruska, 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:

- https://shijuinpallotti.wordpress.com/wp-content/uploads/2019/07/optical-fiber-communications-principles-and-pr.pdf
- https://www.geokniga.org/bookfiles/geokniga-crystallography_0.pdf

- https://dpbck.ac.in/wp-content/uploads/2022/10/Introduction-to-Solid-State-PhysicsCharles-Kittel.pdf
- https://www.thomaswong.net/introduction-to-classical-and-quantum-computing-1e4p.pdf
- https://www.fi.muni.cz/usr/gruska/qbook1.pdf
- https://profmcruz.wordpress.com/wp-content/uploads/2017/08/quantum-computation-and-quantum-information-nielsen-chuang.pdf

ENGLISH FOR SKILL ENHANCEMENT

L	T	P	C
3	0	0	3

Course code: 25HC01
B.Tech I Year I Sem

Course Objectives:

1. Improve their vocabulary.
2. Use appropriate sentence structures in their oral and written communication.
3. Develop their reading and study skills.
4. Equip students to write paragraphs, essays, précis and draft letters.
5. Acquire skills for Technical report writing.

Course Outcomes:

1. Choose appropriate vocabulary in their oral and written communication.
2. Demonstrate their understanding of the rules of functional grammar and sentence structures.
3. Develop comprehension skills from known and unknown passages.
4. Write paragraphs, essays, précis and draft letters.
5. Write abstracts and reports in various contexts.

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 the other aspects of grammar which were not covered in the 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.

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

(Note: As the syllabus of English given in AICTE Model Curriculum-2018 for B.Tech. First

Year is Open-ended, besides following the prescribed textbook, it is required to prepare teaching/learning materials by the teachers collectively in the form of handouts based on the needs of the students in their respective colleges for effective teaching/learning in the class.)

TEXT BOOK:

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

REFERENCES:

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 &amp;PushpLata. 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.

ELECTRONIC DEVICES AND CIRCUITS

Course code: 25CC01
B.Tech I Year I Sem

L	T	P	C
3	0	0	3

Course Objectives:
This course introduces fundamental semiconductor devices and their behavior, including diodes, BJTs, and FETs. It covers their characteristics, applications, and the analysis of basic electronic circuits. The course also explores rectifiers, voltage regulation, amplifier design, and advanced semiconductor technologies like FinFETs and CNTFETs. Emphasis is placed on developing a strong foundation for analog circuit design and understanding modern device technologies in electronics.

Course Outcomes:

1. Analyze the electrical characteristics and models of semiconductor diodes and apply them in rectifier and clipping circuits.
2. Evaluate the operation and configurations of Bipolar Junction Transistors (BJTs) and analyze their input and output characteristics.
3. Design appropriate biasing networks for BJTs and determine the operating point for amplifier applications.
4. Analyze transistor amplifier circuits using h-parameter models and assess performance for various configurations.
5. Analyze the structure, working, and characteristics of JFETs, MOSFETs, and advanced devices like FinFETs and CNTFETs, and compare modern device technologies.

Course Articulation Matrix:

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	2	1	1	-	-	-	-	-
CO2	3	3	2	2	1	-	-	-	-	-	-
CO3	3	3	3	2	1	-	-	-	-	-	-
CO4	3	3	3	2	2	-	-	-	-	-	1
CO5	3	3	2	2	2	1	-	-	-	-	2

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 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, 11 ed., 2013.
3. Sedra, Adel S., and Kenneth C. Smith. Microelectronic Circuits. Oxford University Press, 7th ed., 2014.

REFERENCES:

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.

PROGRAMMING FOR PROBLEM SOLVING

Course code: 25FC01
B.Tech I Year I Sem

L	T	P	C
3	0	0	3

Course Objectives:

1. To learn the fundamentals of computers.
2. To understand the various steps in program development.
3. To learn the syntax and semantics of the C programming language.
4. To learn the usage of structured programming approaches in solving problems.

Course Outcomes:

1. To write algorithms and to draw flowcharts for solving problems.
2. To convert the algorithms/flowcharts to C programs.
3. To code and test a given logic in the C programming language.
4. To decompose a problem into functions and to develop modular reusable code.
5. To use arrays, pointers, strings and structures to write C programs.
6. Searching and sorting problems.

Unit - 1: Introduction to Programming

Compilers, compiling and executing a program.
Representation of Algorithm - Algorithms for finding roots of a quadratic equations, finding minimum and maximum numbers of a given set, finding if a number is prime number Flowchart/Pseudocode with examples, Program design and structured programming

Introduction to C Programming Language: variables (with data types and space requirements), Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, Bitwise operations: Bitwise AND, OR, XOR and NOT operators
I/O: Simple input and output with scanf and printf, formatted I/O

Conditional Branching: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto.

Unit - II: Loops, Functions and Preprocessor Directives:

Loops: Iteration with for, while, do- while loops

Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries

Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions.
Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef

Unit - III: Arrays, Strings, Searching and Sorting:

Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings

Basic searching in an array of elements (linear and binary search techniques), Basic algorithms to sort array of elements (Bubble, Insertion and Selection sort algorithms), Basic concept of order of complexity through the example programs

Unit - IV: Structures, Pointers and Dynamic Memory Allocation

Structures: Defining structures, initializing structures, unions, Array of structures

Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of Pointers in self-referential structures, usage of self referential structures in linked list (no implementation) Enumeration data type
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

Unit - V: File handling in C:

The main method and command line arguments, Introduction to stdin, stdout and stderr. Command line arguments.
Files: Text and Binary files, Creating and Reading and writing text and binary files, appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

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)

REFERENCES:

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, McGrawHill
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

ENVIRONMENTAL SCIENCE

L	T	P	C
1	0	0	1

Course code: 25HC05
B.Tech I Year I Sem

Course Objectives:

1. To understand the importance of ecological balance for sustainable development.
2. Analyze the impact of developmental activities and create mitigation measures.
3. Frame and implement environmental policies and regulations.

Course Outcomes:

1. Demonstrate ecological principles and natural resources. L2
2. Interpret biodiversity and Environmental Pollution. L5
3. Develop technologies on the basis of and environmental regulations which in turn help in sustainable development. L6

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 cycles, Bioaccumulation, Biomagnification, ecosystem value, services and carrying capacity.

UNIT-II Natural Resources: Classification of Resources: Living and Non-Living resources, water resources: use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. Mineral resources: use and exploitation, environmental effects of extracting and using mineral resources, Land Energy resources: growing energy needs, renewable and non renewable energy sources, use of alternate energy source.

UNIT-III Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation.

UNIT-IV Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants. Acid rain-Threshold limit values of chemicals present in environment, Global warming, Ozone layer depletion, Water pollution: Sources and types of pollution. Soil Pollution: Sources and types, Impacts of modern agriculture, degradation of soil. Noise Pollution: Sources and Health hazards, standards, Solid waste: Municipal Solid Waste management, composition and characteristics of e-Waste and its management. Pollution control technologies: Sewage water Treatment, Kyoto protocol, and Montréal Protocol.

UNIT-V Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedical waste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan (EMP).

Towards Sustainable Future: Concept of Sustainable Development Goals, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA), Low carbon life style.

TEXT BOOKS:

1. Perspectives in Environmental Studies: Kaushik A. and Kaushik, C.P. New Age International (P) Ltd. (2008)

REFERENCES:

1. Environmental Studies by Erach Bharucha, 2005 University Press.
2. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
3. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHI Learning Pvt. Ltd.
4. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
5. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
6. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.

ADVANCED ENGINEERING PHYSICS LAB

Course code: 25FC01
B.Tech I Year I Sem

L	T	P	C
0	0	2	1

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 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 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.

Experiments:

1. Synthesis of magnetite (Fe_3O_4) powder using sol-gel method.
2. Determination of energy gap of a semiconductor.
3. Study of a LED characteristics and calculation of its forward resistance.
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 ferroelectric crystal.
7. Determination of dielectric constant of a given material.
8. Determination of Curie's temperature of a given ferroelectric materials.
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 fiber
    (b) Determination of bending losses of a given optical fibre.

Note: Any 8 experiments are to be performed.

ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

Course code: 25HC61
B.Tech I Year I Sem

L	T	P	C
0	0	2	1

The English Language and Communication Skills (ELCS) Lab focuses on listening and speaking skills, particularly on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations both in formal and informal contexts.

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 their pronunciation and neutralize accent
4. To enable students express themselves fluently and appropriately
5. To practise speaking in social and professional contexts

Course Outcomes:

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

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 focuses on speaking skills

The following course content is prescribed for the English Language and Communication Skills Lab.

Exercise – I

CALL Lab:

Instruction: Speech Sounds - Listening Skill - Importance – Purpose - Types - Barriers – Active Listening

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

ICS Lab:

Instruction: The 7 Cs of communication, - formal and informal, verbal and non-verbal

Practice:

Ice breaking Activity (Introducing the partner)

Silent speech – Dumb Charades 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.)

ICSLab:

Instruction: Introduction to English Speech sounds, Errors in Pronunciation – Tips for Neutralizing Mother Tongue Influence (MTI).

Practice:

Minimal Pairs Shadow reading &amp; Reading aloud

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

ICSLab:

Instruction: Enhancing Conversational skills, conversation starters, small talk questions and how to talk to strangers.

Practice:

Conversations with strangers – (contexts - party, seminar, traveling, classroom etc.) Interview each other

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.)

ICSLab:

Instruction: Describing people, objects, places, situations and events.

Practice:

Picture description activity Describing people, objects, places, situations and events

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.)

ICSLab:

Instruction: How to tell a good story – story star- sequencing – creativity

Practice:

Activity on Telling and Retelling stories. - Collage

Reviewing books / movies.

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 audio-visual aids with a Public Address System, a TV or LCD, a digital stereo – audio &amp; video system and camcorder etc.

PROGRAMMING FOR PROBLEM SOLVING LAB

Course code: 25FC61
B.Tech I Year I Sem

L	T	P	C
0	0	2	1

[Note:The programs may be executed using any available Open Source/ Freely available IDE
Some of the Tools available are:
CodeLite: https://codelite.org/
Code::Blocks: http://www.codeblocks.org/
DevCpp :http://www.bloodshed.net/devcpp.html
Eclipse: http://www.eclipse.org
This list is not exhaustive and is NOT in any order of preference]

Course Objectives:

1. To work with an IDE to create, edit, compile, run and debug programs
2. To analyze the various steps in program development.
3. To develop programs to solve basic problems by understanding basic concepts in C like operators, control statements etc.
4. To develop modular, reusable and readable C Programs using the concepts like functions, arrays etc.
5. To write programs using the Dynamic Memory Allocation concept.
6. To create, read from and write to text and binary files

Course Outcomes:

1. formulate the algorithms for simple problems
2. translate given algorithms to a working and correct program
3. correct syntax errors as reported by the compilers
4. identify and correct logical errors encountered during execution
5. represent and manipulate data with arrays, strings and structures
6. use pointers of different types
7. create, read and write to and from simple text and binary files
8. modularize the code with functions so that they can be reused

Cycle 1:

a. Write the program for the simple, compound interest. .
b. Write a simple program that prints the results of all the operators available in C (including pre/ post increment , bitwise and/or/not , etc.). Read required operand values from standard input.
c. Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.
d. Write a 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.
e. Write a C program to find the roots of a Quadratic equation.
f. Write a program for finding the max and min from the three numbers
g. 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)

Cycle 2:

a. 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
```

b. Write a program that shows the binary equivalent of a given positive number between 0 to 255.

c. A building has 10 floors with a floor height of 3 meters each. A ball is dropped from the top of the building. Find the time taken by the ball to reach each floor. (Use the formula s = ut+(1/2)at^2 where u and a are the initial velocity in m/sec (= 0) and acceleration in m/sec^2 (= 9.8 m/s^2)).

d. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.

e. Write a program that finds if a given number is a prime number

f. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

Cycle 3:

a. Write a C program to calculate the following, where x is a fractional value.
1-x/2 +x^2/4-x^3/6

b. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

c. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: 1+x+x^2+x^3+………….+x^n. For example: if n is 3 and x is 5, then the program computes 1+5+25+125.

d. Write a C program to construct a pyramid of numbers as follows:

```
1             *             1
1 2           * *           2 3
1 2 3         * * *         4 5 6
```

Cycle 4:

a. Write C programs that use both recursive and non-recursive functions To find the factorial of a given integer.
To find the GCD (greatest common divisor) of two given integers. To find x^n

b. Write a menu driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average. The menu and all the choices are to be functions. Use a switch statement to determine what action to take. Display an error message if an invalid choice is entered.

Cycle 5:

a. Write a C program to find the minimum, maximum and average in an array of integers.

b. Write a function to compute mean, variance, Standard Deviation, sorting of n elements in a single dimension array.

c. Write a C program that uses functions to perform the following:

Addition of Two Matrices
Multiplication of Two Matrices

d. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.

Cycle 6:

a. 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.)
b. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn‘t containch.
c. Write a C program to count the lines, words and characters in a given text.
d. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
e. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
f. Write a C program that uses functions to perform the following operations:
To insert a sub-string into a given main string from a given position.
To delete n Characters from a given position in a given string.

Cycle 7:

a. Write a C program that uses non recursive function to search for a Key value in a given
list of integers using linear search method.
b. Write a C program that uses non recursive function to search for a Key value in a given
sorted list of integers using binary search method.
c. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.

Cycle 8:

a. Write a C program that sorts the given array of integers using selection sort in descending order
b. Write a C program that sorts the given array of integers using insertion sort in ascending order
c. Write a C program that sorts a given array of names

Cycle 9:

a. Write a program for reading elements using a pointer into an array and display the values using the array.
b. Write a program for display values reverse order from an array using a pointer.
c. Write a program through a pointer variable to sum of n elements from an array.

Cycle 10:

a. Write a C program to display the contents of a file to standard output device.
b. Write a C program which copies one file to another, replacing all lowercase characters with their uppercase equivalents.

c. Write a C program to count the number of times a character occurs in a text file. The file name and the character are supplied as command line arguments.

Cycle 11:

a. Write a C program that does the following:
   a. It should first create a binary file and store 10 integers, where the file name and 10 values are given in the command line. (hint: convert the strings using atoi function)
   b. Now the program asks for an index and a value from the user and the value at that index should be changed to the new value in the file. (hint: use fseek function)
   c. The program should then read all 10 values and print them back.

b. 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).

Cycle 12: CASE STUDY

Student Database Management System Using Flat Files.(Binary Mode File Organization using fwrite and fread methods)
Student Record should consists of fields RollNo(int), Name(string of max length 30), Gender(char), Percentage(float)

Constraints: 1. No duplicates on RollNo
2. Records must be arranged automatically in ascending order of their RollNo
3. Percentage must be in between 0 and 100

Operations to be implemented:

1. Add new student details
2. Search Student Details
   a. By their RollNo
   b. By Percentage>= given percentage
3. Update Student Details based on their RollNo(Only Name and Percentage are allowed to be modified)
4. Delete Student Record based on their RollNo
5. Display All the Students Details

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)

REFERENCES:

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, McGrawHill
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

ENGINEERING WORKSHOP

Code: 25BC61
B. Tech. I Year I Semester
Pre-requisites: Practical skill

L	T	P	C
0	0	2	1

Course Objectives:

1. To Study of different hand operated power tools, uses and their demonstration.
2. To gain a good basic working knowledge required for the production of various engineering products.
3. To provide hands on experience about use of different engineering materials, tools, equipments and processes those are common in the engineering field.
4. To develop a right attitude, team working, precision and safety at work place.
5. It explains the construction, function, use and application of different working tools, equipment and machines.
6. To study commonly used carpentry joints.
7. To have practical exposure to various welding and joining processes.
8. Identify and use marking out tools, hand tools, measuring equipment and to work to prescribed tolerances.

Course Outcomes:

1. Study and practice on machine tools and their operations
2. Practice on manufacturing of components using workshop trades including pluming, fitting, carpentry, foundry, house wiring and welding.
3. Identify and apply suitable tools for different trades of Engineering processes including drilling, material removing, measuring, chiseling.
4. Apply basic electrical engineering knowledge for house wiring practice.

1. TRADES FOR EXERCISES:

At least two exercises from each trade:

I. Carpentry – (T-Lap Joint, Dovetail Joint, Mortise &amp; Tenon Joint)

II. Fitting – (V-Fit, Dovetail Fit &amp; Semi-circular fit)

III. Tin-Smithy – (Square Tin, Rectangular Tray &amp; Conical Funnel)

IV. Foundry – (Preparation of Green Sand Mould using Single Piece and Split Pattern)

V. Welding Practice – (Arc Welding &amp; Gas Welding)

VI. House-wiring – (Parallel &amp; Series, Two-way Switch and Tube Light)

2. TRADES FOR DEMONSTRATION &amp; EXPOSURE:
Machine Shop, Power tools in construction and Wood Working

TEXT BOOKS:

1. Workshop Practice /B. L. Juneja / Cengage
2. Workshop Manual / K. Venugopal / Anuradha.

REFERENCES:

1. Work shop Manual - P. Kannaiah/ K.L. Narayana/ Scitech
2. Workshop Manual / Venkat Reddy/ BSP

Syllabus of I Year II Sem

ORDINARY DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS

(Common to all Branches of Engineering)

Course Code: 25HC08

B.Tech I Year II Sem

Pre Requisites: Mathematics Knowledge at Pre-University Level

L	T	P	C
3	0	0	3

Course Objectives:

1. Analytical solutions of ODE.
2. Laplace Transform technique
3. Vector calculus

Course Outcomes:

Find the solutions of first order first degree differential equations and solve the problems on Newton’s law of cooling, Natural growth and decay.

1. Solve higher order ordinary differential equations with constant coefficients using some standard methods
2. Evaluate the Laplace transform of a given function and find Inverse Laplace transform, also solve initial value problems.
3. Find directional derivative and solve the problems on line, surface and volume integrals.

UNIT-I: Ordinary Differential Equations of First Order: Exact differential equations, Equations reducible to exact, Linear and Bernoulli’s equations, Orthogonal Trajectories (only in Cartesian Coordinates). Applications: Newton’s law of cooling, Law of natural growth and decay. [10 hours]

UNIT- II: Ordinary Differential Equations of Higher Order: Higher order linear differential equations with constant coefficients-Complementary function, Particular Integral**.** Non-Homogeneous terms of the type e^{ax}, sinax, cosax, polynomials in x, e^{ax} .V(x) and x. V(x), Method of variation of parameters. [10 hours]

UNIT-III: Laplace Transform: Laplace transform of standard functions, shifting theorems, change of scale property, Laplace Transform of derivatives and integrals, multiplication by powers of ‘t’, division by ‘t’ (without proofs), Evaluation of integrals by Laplace transforms.

Inverse Laplace Transform: Properties, Inverse Laplace Transform by partial fraction method and convolution theorem (without proof). Application: Solving initial value problems by Laplace Transforms. [10 hours]

UNIT-IV: Vector Differentiation: Vector and scalar point functions. Gradient of scalar function and its physical interpretation, Angle between the two surfaces, Directional derivatives; Divergence and Curl of a vector point function, Solenoidal vectors and Irrotational vectors, Scalar potential functions, Laplacian Operator, Vector Identities (without proofs).

UNIT-V: Vector Integration: Line integral, circulation, Work done by force vector, Evaluation of Surface and Volume Integrals. Vector integral theorems: Green’s theorem, Gauss-Divergence theorem, Stoke’s theorem (without proofs) and their problems.

TEXT BOOKS:

1. R K Jain and S R K Iyengar Advanced Engineering Mathematics, Narosa Publications.

2. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers.
3. Alan Jeffery, Advanced Engineering Mathematics, Academic Press
4. B. V. Ramana, Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
5. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.

ENGINEERING CHEMISTRY
(Common to all branches)

Course Code: 25HC04
B.Tech I Year II Sem

L	T	P	C
3	0	0	3

Course Objectives:

1. To bring adaptability to new developments in Engineering Chemistry and to acquire the skills required to become a perfect engineer.
2. To include the importance of water in industrial usage, fundamental aspects of battery chemistry, significance of corrosion it's control to protect the structures.
3. To imbibe the basic concepts of polymer and their applications.
4. To acquire required knowledge about engineering materials.

Course Outcomes:

1. Outline basic properties of water and its usage in domestic and industrial purposes[L2]
2. Summarize electrochemical procedures related to corrosion and energy sources[L2]
3. Interpret potential applications of polymers and engineering materials [L5]

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 break-point 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. Construction, working and determination of pH of an unkown solution using SHE and 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, Cracking - Types of cracking - Moving bed catalytic cracking. LPG and CNG composition and uses.
Synthetic Fuels: 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 themoplastics and thermo setting plastics, Fiber reinforced plastics (FRP).
Conducting polymers: Definition and Classification with examples - Mechanism of conduction in trans-poly-acetylene and applications of conducting polymers.
Biodegradable polymers: Polylactic acid and its applications.

UNIT-V- Materials Chemistry:
Lubricants: Definition and characteristics of a good lubricant – thin film mechanism of lubrication, properties of lubricants - viscosity, cloud and pour point, flash and fire point.
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 Under Control- CO sensor (Passive Infrared detection), Raman spectroscopy (application) - Tumour detection in medical applications.

TEXT BOOKS:

1. Engineering Chemistry: PK Jain & MK Jain, Dhanapathrai Publications (2018)
2. Engineering Chemistry: by Thirumala Chary Laxminarayana & Shashikala, Pearson Publications (2020)

REFERENCES:

1. Textbook of Engineering Chemistry: Jaya Shree Anireddy, Wiley Publications (2019)
2. Engineering Chemistry: by & B.Rama Devi, Prasanta Rath & Ch. Venkata Ramana Reddy, Cengage Publications (2018)
3. Engineering Chemistry: Shashi Chawla, Dhanapathrai Publications (2019)
4. Textbook of Engineering Chemistry: SS Dara, SS Umare S. Chand Publications (2004)

ENGINEERING DRAWING AND COMPUTER AIDED DRAFTING

(COMMON TO ALL BRANCHES)

L	T	P	C
2	0	2	3

Course Code: 25BC01
B.Tech I Year II Sem
Pre Requisites: Nil

Course Objectives:

1. Engineering Drawing is the language of Engineers that meant for communicating one's ideas, thoughts, and designs to others. A Drawing drawn by an engineer, having sound engineering knowledge, for the Engineering purposes is an Engineering Drawing. Drawing is a starting point for all engineering branches.
2. The main objective is to read and communicate developments in engineering field through the engineering drawing whatsoever branch he/she may belong to.
3. Even a small error committed by an Engineer in his/her drawing may lead to misguiding the workman/operator and the ultimate product/construction will be a flop. Every language has its own rules of grammar.

Course outcomes:

1. Outline the basics of engineering graphics, engineering curves and Auto CAD
2. Interpret the orthographic projections of points, lines, Planes, and solids.
3. Draw the Sections of Solids
4. Develop Isometric views and conversions from isometric views to orthographic.

UNIT – I

Introduction to Engineering Drawing: Drawing Instruments and their uses, types of lines, Lettering, Dimensioning-Terms & notations, placing of dimensions, general rules of dimensioning **Scales**(concepts).**:** RF, Reducing, Enlarging and Full Scales

Curves: Conic Sections including Rectangular Hyperbola - General method, Cycloid, and Involutes of circle.

Introduction to CAD: Benefits of CAD, Graphic input, and output devices - Function performed by CAD Software, AUTOCAD-Drawing Entities, Editing commands.

UNIT – II

Orthographic Projection: Principles of Projection – Methods of projection, First angle and third angle projections.

Projections of Points & Lines: Projections of Points, Projections of straight lines –line inclined to one plane and line inclined to both reference planes.

UNIT –III

Projections of regular Planes: types of planes, Projections of plane inclined to one reference plane and Oblique planes.

UNIT – IV

Projections of regular Solids: types of solids, Projections of: Prisms, Cylinders, Pyramids, Cones – simple position and axis inclined to one plane only

Sections and sectional views of Solids: Sections and Sectional views of Right Regular Solids – Prism, Pyramid – Auxiliary views.

UNIT – V

Isometric Projection: Introduction, Isometric axes, lines and planes, Isometric Scale – Isometric drawing or View – Isometric drawing of planes and simple solids such as prisms, pyramids, cylinder, cone.

Conversions: Conversion of isometric views to orthographic views of simple objects.

TEXT BOOKS:

1. Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing House (In First-angle Projection Method)

REFERENCES:

1. Agrawal B. &Agrawal C. M. (2012), Engineering Graphics, TMH Publication
2. AUTOCAD Software Theory and User Manuals

BASIC ELECTRICAL ENGINEERING

L	T	P	C
3	0	0	3

Course Code: 25AC47
B.Tech I Year II Sem
Prerequisites: Mathematics

Course Objectives:

1 To understand DC and Single & Three phase AC circuits
2 To study and understand the different types of DC, AC machines and Transformers.
3 To import the knowledge of various electrical installations and the concept of power, power factor and its improvement.

Course Outcomes:

1 Understand and analyze basic Electrical circuits
2 Study the working principles of Electrical Machines and Transformers
3 Introduce components of Low Voltage Electrical Installations.

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.

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 and of dc machine, Construction and working principle of a three-phase induction motor, Construction and working principle of a Single-phase induction motor, Construction and working principle of a synchronous machines (generator and motor), Applications of all machines.

UNIT-V:

Electrical Installations: Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, ELCB, MCCB, 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.

REFERENCES:

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

DATA STRUCTURES

L	T	P	C
3	0	0	3

Course Code: 25EC01
B.Tech I Year II Sem
Prerequisites: A course on "Programming for Problem Solving

Course Objectives:

1. Exploring basic data structures such as stacks and queues.
2. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
3. Introduces sorting and pattern matching algorithms.

Course Outcomes:

1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.

UNIT – I

Introduction to Data Structures: Basic Terminology, Classification of Data Structures, Operation on Data Structures, abstract data types, selecting a Data Structure, Linear list – Introduction, singly linked list, Circular Linked Lists, Doubly Linked List, Stacks-Operations, Stack algorithm, Stack ADT, Stack applications, Queues- operations, Queue Algorithm, Queue ADT, Queue Applications.

UNIT - II

Trees: Introduction, Types of Trees, creating a Binary Tree from a General Tree, traversing a Binary Tree, Binary Search Trees (BST), BST Operations- Searching, Insertion and Deletion, BST ADT, BST Applications, Threaded Binary Trees, AVL Trees, Red –Black Trees, Splay Trees

UNIT – III

Multi way Search Trees: Introduction, B Trees, B Trees ADT, 2-3 Trees, 2-3- Tree, B* Tree, B+ Trees

Heaps: Binary Heaps, Binomial heaps, Fibonacci heaps, Comparison of Various Heaps, Applications

Searching: Introduction, Interpolation Search, Jump search

UNIT - IV

Graphs: Introduction, Directed Graphs, Bi connected Components, Representation of Graphs, Graph Traversal Algorithms, Graph ADT, Applications of Graphs

Sorting: Radix Sort, Heap sort, Shell Sort, Tree Sort,

UNIT – V

Hashing and Collision: Introduction, Hash Tables, Hash Functions, Different Hash Functions: Division Method, Multiplication Method, Mid-square Method, Folding Method; collisions: Collision Resolution by Open Addressing, Collision Resolution by Chaining
Files and their Organization: Introduction, Data hierarchy, File Attributes, Text and Binary Files, Basic File Operations, File Organization, Indexing

TEXTBOOKS:

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

REFERENCES:

1. Data Structures using C – A. S.Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson Education.

ENGINEERING CHEMISTRY LAB
(Common to all branches)

Course Code: 25HC65
B.Tech I Year II Sem

L	T	P	C
0	0	2	1

Course Objectives:

1. Estimation of hardness of water to check its suitability for drinking purpose.
2. Students are able to perform estimations of acids and bases using conductometry, potentiometry.
3. Students will learn to prepare organic compound, polymers such as Aspirin, Bakelite and nylon in the laboratory.
4. Students will learn skills related to the lubricant properties such viscosity of oils.

Course Outcomes:

1. Estimate the hardness in water. L5
2. Determine strength of acid by potentiomeric and conductometric methods. L5
3. Determine viscosity of oil. L5
4. Demonstrate preparation of polymer, aspirin and inorganic compound. L2

List of Experiments

I. **Volumetric Analysis:** Estimation of Hardness of water by EDTA Complexometry method.

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.

III. **Potentiometry:**
1. Estimation of concentration of Fe^{+2}ion by Potentiometry using $KMnO_4$.
2. Estimation of concentration of strong acid with strong base by Potentiometry using quinhydrone

IV. **Determination of Viscosity.**

V. **Colorimetry:** Verification of Lambert-Beer's law using $KMnO_4$.

VI. **Preparations:**
1. Preparation of Bakelite.
2. Preparation Nylon – 6, 6.
3. Preparation of Aspirin

VII. **Corrosion:** Determination of rate of corrosion of mild steel in the presence and absence of inhibitor.

VIII. **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.

REFERENCES:

1. Vogel’s text book of practical organic chemistry 5th edition
2. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.
3. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007).

BASIC ELECTRICAL ENGINEERING LAB

Course Code: 25AC77
B.Tech I Year II Sem
Prerequisites: Basic Electrical Engineering

L	T	P	C
0	0	2	1

Course Objectives:

1. To measure the electrical parameters for different types of DC and AC circuits using conventional and theorems approach.
2. To study the transient response of various R, L and C circuits using different excitations.
3. To determine the performance of different types of DC, AC machines and Transformers.

Course Outcomes:

1. Verify the basic Electrical circuits through different experiments.
2. Evaluate the performance calculations of Electrical Machines and Transformers through various testing methods.
3. Analyze the transient responses of R, L and C circuits for different input conditions.

List of experiments/demonstrations:

PART- A (compulsory)

1. Verification of KVL and KCL
2. Verification of Thevenin’s and Norton’s theorem
3. Resonance in series RLC circuit
4. Calculations and Verification of Impedance and Current of RL, RC and RLC series circuits
5. Measurement of Voltage, Current and Real Power in primary and Secondary Circuits of a Single-Phase Transformer
6. Performance Characteristics of a DC Shunt Motor

PART-B (any 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

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.

DATA STRUCTURES LAB

L	T	P	C
0	0	2	1

Course Code: 25EC61
B.Tech I Year II Sem
Prerequisites: A Course on "Programming for problem solving".

Course Objectives:

1. It covers various concepts of C programming language
2. It introduces searching and sorting algorithms
3. It provides an understanding of data structures such as stacks and queues.

Course Outcomes:

1. Ability to develop C programs for computing and real-life applications using basic elements like control statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
2. Ability to Implement searching and sorting algorithms

List of Experiments

1. Write a program that uses functions to perform the following operations on singly linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
3. Write a program that uses functions to perform the following operations on circular linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
4. Write a program that implement stack (its operations) using
   i) Arrays ii) ADT
5. Write a program that implement Queue (its operations) using
   i) Arrays ii) ADT
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order
   i) Radix Sort, ii) Heap sort, iii) Shell Sort, iv) Tree Sort
7. Write a program to implement the tree traversal methods (Recursive and Non-Recursive).
8. Write a program to implement
   i) Binary Search tree ii) B Trees iii) B+ Trees iv) AVL trees v) Red - Black trees
9. Write a program to implement the graph traversal methods.
10. Write a program to implement the following Hash Functions: i) Division Method, ii) Multiplication Method, iii) Mid-square Method, iv) Folding Method

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.

REFERENCES:

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

PYTHON PROGRAMMING LAB

Course Code: 25FC61
B.Tech I Year II Sem

L	T	P	C
0	0	2	1

Course Objectives:

1. To install and run the Python interpreter
2. To learn control structures.
3. To Understand Lists, Dictionaries in python
4. To Handle Strings and Files in Python

Course Outcomes:

1. Develop the application specific codes using python.
2. Understand Strings, Lists, Tuples and Dictionaries in Python
3. Verify programs using modular approach, file I/O, Python standard library
4. Implement Digital Systems using Python

Note: The lab experiments will be like the following experiment examples

Week -1:

1. i) Use a web browser to go to the Python website http://python.org. This page contains information about Python and links to Python-related pages, and it gives you the ability to search the Python documentation.
   ii). Start the Python interpreter and type help() to start the online help utility.
2. Start a Python interpreter and use it as a Calculator.
3. i)Write a program to calculate compound interest when principal, rate and number of periods are given.
   ii). Given coordinates (x1, y1), (x2, y2) find the distance between two points
4. Read the name, address, email and phone number of a person through the keyboard and print the details.

Week-2:

1. Print the below triangle using for loop.
   5
   4 4
   3 3 3
   2 2 2 2
   1 1 1 1 1
2. Write a program to check whether the given input is digit or lowercase character or uppercase character or a special character(use 'if-else-if' ladder)
3. Python Program to Print the Fibonacci sequence using while loop
4. Python program to print all prime numbers in a given interval (use break)

Week -3:

1. Write a program to convert a list and tuple into arrays.
2. Write a function called gcd that takes parameters a and b and returns their greatest common divisor.

3. Write a function called palindrome that takes a string argument and returnsTrue if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

Week-4:

1. Write a function called is_sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise.
2. Write a function called has_duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.
3. Write a function called remove_duplicates that takes a list and returns a new list with only the unique elements from the original. Hint: they don't have to be in the same order.

Week-5:

1. Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
2. Write a python code to read dictionary values from the user. Construct a function to invert its content. i.e., keys should be values and values should be keys.
3. Write a recursive function that generates all binary strings of n-bit length.
4. Write a recursive function that generates factorial of a given number.

Week-6:

1. Write a python program that defines a matrix and prints the matrix.
2. Write a python program to perform addition of two square matrices
3. Write a python program to perform multiplication of two square matrices

Week-7:

1. Demonstrate different types of inheritance like Single Inheritance, Multi level Inheritance.
2. Write a Python program to demonstrate the usage of Method Resolution Order (MRO) in multiple levels of Inheritances.
3. Write a python code to read a phone number and email-id from the user and validate it for correctness.

Week- 8

1. Write a Python code to merge two given file contents into a third file.
2. Write a Python code to open a given file and construct a function to check for given words present in it and display on found.

Week-9:

1. Write a Python code to Read text from a text file, find the word with most number of occurrences
2. Write a function that reads a file *file1*and displays the number of words, number of vowels, blank spaces, lower case letters and uppercase letters.

Week-10:

1. Install NumPypackage with pip, Import numpy and explore it.
2. Install Plotpy and Scipy and explore their functionalities.

Week-11:

1. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
2. Write a program to implement Half Adder, Full Adder.

Week-12:

1. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

TEXTBOOKS:

1. Supercharged Python: Take your code to the next level, Overland
2. Learning Python, Mark Lutz, O'reilly

REFERENCES:

1. Python Programming: A Modern Approach, VamsiKurama, Pearson
2. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, SheetalTaneja, Naveen Kumar, Pearson
3. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
4. Python for Data Science, Dr. Mohd Abdul Hameed, wiley publications
5. Core Python Programming, Dr.R.Nageswara Rao, dreamtech press
6. Introduction to Python, Gowrishankar S, Veena A., CRC Press

IT WORKSHOP

Course Code: 25FC63
B.Tech I Year II Sem

L	T	P	C
0	0	2	1

Course Objectives:

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. Document/ Presentation preparation
4. Perform calculations using spreadsheets

PC Hardware

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. 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 2: 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 3: 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 installation and follow it up with a Viva

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 Word 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

Power point

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 PowerPoint.
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.

HTML, CSS:

Task 1: Students will be working on the basics of HTML using all the tags.
Task 2: Creating HTML with CSS
Task 3: Creating HTML with CSS using Bootstrap

REFERENCES:

1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
2. The Complete Computer upgrade and repair book, 3rd edition Cheryl A Schmidt, *WILEY Dreamtech*
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.*

Syllabus of II Year I Sem

DISCRETE MATHEMATICS

Course Code: 25FC02
B.Tech II Year I Sem

L	T	P	C
3	0	0	3

Course Objectives:

1. Introduces elementary discrete mathematics for computer science and engineering.
2. Topics include 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:

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

UNIT - I

Mathematical logic: Statement and notations, Connectives, Well-formed Formulas, Truth Tables, Tautology, Equivalence, Implication, Rules of inference, Arguments, Proof by contradiction, Conditional Proof Normal forms, and Automatic theorem proving, Predicates: Quantifiers, Free and Bound variables, Rules of inference, Consistency

UNIT - II

Recurrence relations: Generating functions. Function of sequences. Recurrence relations, Solving recurrence relations by substitution and generating functions. Characteristic roots. Solution of Inhomogeneous recurrence relations

UNIT - III

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures,

Relations: Properties of Binary Relations, Equivalence, transitive closure, Compatibility & Partial Ordering Relations, Hasse Diagrams, Lattice and its properties, Functions.

Algebraic structures: Algebraic systems, Examples and general properties, Semi groups and Monoids. Groups, Subgroups, Ring, Homomorphism, Isomorphism.

UNIT - IV

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

UNIT - V

Graph Theory: Basic Concepts, Isomorphism and Sub graphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs 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 ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe l. Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCES:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill publishing co.

COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code: 25CC47
B.Tech II Year I Sem

L	T	P	C
3	0	0	3

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, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors

Course Outcomes:

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:

Boolean Algebra and Logic Gates: Binary codes, Binary Storage and Registers, Binary logic.
Digital logic gates. Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation
Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

UNIT - II:

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers
Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure. Registers, shift Registers, Ripple counters, synchronous counters, other counters.

UNIT III

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 - IV

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.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT - V

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.

TEXT BOOKS:

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

REFERENCES:

1. Switching and Finite Automata Theory, ZVI. Kohavi, Tata McGraw Hill.
2. Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, 5th Edition, McGrawHill.
3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
4. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Course Code: 25EC02
B.Tech II Year I Sem

L	T	P	C
3	0	0	3

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:

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.util package, 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.

UNIT - V

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, JCheckBoxMenuItem, JRadioButtonMenuItem, JSeperator. creating a popup menu

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.

REFERENCES:

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.

SOFTWARE ENGINEERING

Course Code: 25EC03

B.Tech II Year I Sem

L	T	P	C
3	0	0	3

Course Objectives:

- 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.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes:

- Ability to translate end-user requirements into system and software requirements, using e.g.
- UML, and structure the requirements in a Software Requirements Document (SRD).
- 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.
- 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, Incremental Process Models, Concurrent Models, Component based development and Agile Development.

UNIT - II

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

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

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model.

Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams and 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.
3. The unified modeling language user guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

REFERENCES:

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.
4. Fundamentals of Software Engineering-Rajib Mall, PHI.

DATABASE MANAGEMENT SYSTEMS

Course Code: 25FC03
B.Tech II Year I Sem
Prerequisites: Data Structures

L	T	P	C
3	0	0	3

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:

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 Models, 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, Comparison Operators, Set operators, Nested Queries, aggregation operators, Joins, 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

REFERENCES:

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.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Course Code: 25EC62
B.Tech II Year I Sem

L	T	P	C
0	0	2	1

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:

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.

Note:

1. Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the use of the Eclipse platform.
2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

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 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.

3.
   A) Develop an applet in Java that displays a simple message.
   B) Develop an applet in Java that receives an integer in one text field, and computes its factorial

4. Value and returns it in another text field, when the button named “Compute” is clicked.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.
11. Write a java program to display the table using Labels in Grid Layout.

12. 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).

13. 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).

14. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.

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.

REFERENCES:

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.

SOFTWARE ENGINEERING LAB

Course Code: 25EC63
B.Tech II Year I Sem

L	T	P	C
0	0	2	1

Prerequisites: Programming for Problem Solving, Software Engineering

Course Objectives:
To have hands-on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

Course Outcomes:

1. Ability to translate end-user requirements into system and software requirements
2. Ability to generate a high-level design of the system from the software requirements
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments
Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

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.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCES:

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill

DATABASE MANAGEMENT SYSTEMS LAB

Course Code: 25FC64
B.Tech II Year I Sem

L	T	P	C
0	0	2	1

Course Objectives:

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

Course Outcomes:

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
   a. Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)
      B) Nested, Correlated subqueries
6. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
7. Triggers (Creation of insert trigger, delete trigger, update trigger)
8. Procedures
9. 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.

REFERENCES:

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.

UNIVERSAL HUMAN VALUES
(Common to All)

Course Code: 25HC06
B.Tech II Year I Sem

L	T	P	C
1	0	0	1

Course Objectives:

1. Develop a holistic perspective towards life through self-exploration, with a focus on understanding the human being, family, society, and nature/existence.
2. Gain clarity on the harmony that exists within the individual and among human relationships, social systems, and natural ecosystems.
3. Strengthen the capacity for self-reflection, encouraging critical introspection and self-awareness.
4. Foster commitment and courage to act based on right understanding and values, to contribute meaningfully towards an ethical and harmonious society.

Unit 1: Introduction

1. Introduction to Values – Definition, Importance, and the Need for Value-Based Education (with reference to values, skills and human goals).
2. Understanding Basic Human Aspirations – Happiness, Continuous Happiness, and Prosperity.
3. Self-Exploration – Purpose, Process, and Natural Acceptance.
4. Understanding Right Understanding, Relationship, and Physical Facility – Distinguishing Human and Animal Consciousness.

Unit 2: Harmony in the Human Being – At Level-1- Individual

5. Understanding the Human Being as a Co-existence of **'I'** and the **'Body'**.
6. Exploring the Needs, Characteristics, and Activities of **'I'** and the **'Body'**.
7. **'Body'** as an Instrument of **'I'** – Role of **'I'** as the Doer, Seer, and Enjoyer.
8. Harmony of **'I'** with the **'Body'** – Through Health (Swasthya), Self-regulation (Sanyam), and Well-being.

Unit 3: Harmony in the Human Being – At Level-2 & 3- Family and Society

9. Universal Human Values in Relationships – The 9 Core Values.
10. Understanding Trust and Respect – Difference between Intention and Competence; Basis for Value-based Differentiation.
11. Harmony in Society – Through Resolution, Prosperity, and Fearlessness.
12. Comprehensive Human Goals – Integration of 5 Dimensions of Human Endeavour and the Concept of "From Family to World Family."

Unit 4: Harmony in the Human Being – At Level-4- Nature

13. Understanding Nature – Definition and the Four Orders of Nature.
14. Mutual Fulfillment and Interconnectedness among the Four Orders.
15. Principles of Recyclability and Self-Regulation in Nature.
16. Existence as Co-existence – Interacting Units in All-pervasive Space.

Unit 5: Holistic Understanding and its Implications

17. Humanistic Constitution and Humanistic Universal Order
18. Holistic Production and Technology Development
19. Holistic Education and Humanistic Management
20. Case Studies – For Self-study and Exploration.
21. Ethics, Morals, Values, and Law – A Critical Examination.
22. Professions and Businesses – Ethical Foundations versus Economic Orientation.
23. Ethical Dilemmas and Clarity on Ethical Human Conduct.
24. Case Studies – For Self-study and Practice; Summary and Reflection.

TEXT BOOKS:

1. Human Values and Professional Ethics by R R Gaur, R Sangal, G P Bagaria, Excel Books, New Delhi, 2010.
2. Business Ethics- An Indian Perspective by Ronald Francis & Mukti Mishra, TMH, New Delhi, 2009

REFERENCES:

1. Jeevan Vidya: EkParichaya, A. Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999. Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
2. The Story of Stuff (Book).
3. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
4. Small is Beautiful - E. F Schumacher.
5. Slow is Beautiful - Cecile Andrews
6. Economy of Permanence - J C Kumarappa
7. Bharat Mein Angreji Raj – Pandit Sunderlal
8. Rediscovering India - by Dharampal
9. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
10. India Wins Freedom - Maulana Abdul Kalam Azad
11. Vivekananda - Romain Rolland (English)
12. Gandhi - Romain Rolland (English)

CODING SKILLS

L	T	P	C
0	0	2	1

Course Code: 25EC04
Prerequisites: Programming for Problem Solving, Data Structures

Course Objectives:

1. To implement programs for real time applications.
2. To enhance problem solving skills and coding skills
3. To develop programs using appropriate data structure.
4. To perform data base management operations using flat files.

Course Outcomes:

1. Build custom code for solving real time applications.
2. Demonstrate various data structures and their usage.
3. Design and develop applications using files.
4. Develop applications using appropriate data structures.

Exercises:

1. Sudoku Validator

Topic: Arrays

Description:

Write a C program to validate a given 9x9 Sudoku grid. Your program should check whether the grid satisfies Sudoku rules: each row, column, and 3x3 subgrid contains all digits from 1 to 9 exactly once. The program should output whether the Sudoku grid is valid or not.

2. Prime Number Matrix

Topic: Arrays

Description:

Write a C program that generates a 5x5 matrix filled with random integers (range 1 to 100). Replace all non-prime numbers with zero. Finally, display the matrix and count the total prime numbers present.

3. Custom String Manipulation Toolkit

Topic: Strings

Description:

Create a C program that implements the following string manipulation functions manually (without using

<string.h> functions):

Calculate string length

Copy a string

Reverse a string

Convert a string to uppercase

Concatenate two strings (user-defined function)

The program should allow the user to input strings and demonstrate all these operations.

4. Enhanced Sentence Analyzer

Topic: Strings

Description:

Develop a C program to analyze a user-input sentence. Your program should:

Count total words

Count vowels, consonants, digits, and special characters

Identify the longest and shortest words

Replace all spaces with hyphens and display the modified sentence

All operations must be implemented manually without using built-in string library functions.

5. Big Number Addition

Topic: Strings + Manual Arithmetic

Description:

Write a C program that accepts two very large numbers as strings (up to 1000 digits) and computes their sum. Implement the addition manually by processing the numbers digit-by-digit without using any built-in big integer libraries or data types. Display the resulting sum as a string.

6. Dynamic Contact Book

Topic: Dynamic Memory Allocation

Description:

Create a C program to manage a contact book dynamically. The program should allow the user to:

Add new contacts (name and phone number) dynamically using malloc and realloc

View all stored contacts

Exit the program while properly freeing all allocated memory

The program should use dynamic memory allocation for storing contacts and their details.

7. Employee Record System

Topic: File Handling

Description:

Develop a C program to manage employee records stored in a text file. Implement functionalities to add new employee data (ID, name, department, salary), view all employee records, and search for employees by ID. The program should use file operations (fopen, fread, fwrite, etc.) and handle all data persistently.

8. Train Route Management System using Files

Topic: File Handling

Description:

A railway reservation system needs to maintain and manage records of various trains and their route details. Instead of using in-memory structures like linked lists, all train data must be stored and retrieved from a flat file.

You are required to implement a Train Route Management System in C language that performs key operations such as adding train details, searching train records based on source or destination code, and displaying all train records — all using file handling techniques.

Train Record Fields:

Each train record must include the following:

Train Number (integer)

Source Code (string)

Source Name (string)

Destination Code (string)

Destination Name (string)

Distance (km) (float/int)

Arrival Time (string, e.g., "14:45")

Departure Time (string, e.g., "15:00")

File Used: All records are to be stored in a flat file named: trains_data.txt

Each record can be read/stored in binary format using fread and fwrite functions.

9. Implement Browser Back Button Using Stack (Simulated with Queue)

Topic: Stack and Queue Data Structures

Description:

In web browsers, the Back button allows users to return to the previous page — this is a classic example of LIFO (Last In, First Out) behaviour, best modelled using a stack.

You are required to implement the Back Button functionality for a simplified web browser. However, you are restricted to using only a queue for the internal data structure.

10. Implement a Hybrid Sorting Algorithm (Merge Sort + Insertion Sort)

Topic: Sorting Techniques

Description:

In many practical applications, using a single sorting algorithm may not always provide the best performance.

Your task is to implement a hybrid sorting algorithm that:

Uses Insertion Sort for small subarrays (below a given threshold)

Uses Merge Sort for larger subarrays

Applies recursion smartly based on subarray size

11. Task Scheduling Using a Priority Queue

Topic: Priority Queue

Description:

You are tasked with building a Task Scheduling System that schedules and executes tasks based on their priority.

Each task has:

A Task ID

A Description

A Priority Level (lower number = higher priority)

You must implement a priority queue to manage these tasks such that:

Tasks with higher priority (lower number) are always processed before others.

If two tasks have the same priority, the one inserted earlier is processed first (FIFO for equal priority).

12. Detect Cycle in a Linked List

Topic: Single Linked List

Description:

You are working as a software engineer at a logistics company that tracks the delivery path of parcels. Each parcel's delivery route is stored as a linked list, where each node represents a location with a unique ID. Occasionally, due to GPS errors or wrong entries, the route forms a loop, indicating that the parcel is stuck in a cycle of locations.

Your task is to write a function to detect whether the delivery route contains a cycle, i.e., if the parcel visits a location more than once by looping through previous locations.

You are provided with the head of the linked list representing the parcel's delivery route.

13. Student Database Management using Binary Search Tree (BST)

Topic: Trees

Description:

You are tasked with developing a Student Database Management System using a Binary Search Tree (BST) in C. Each student record consists of:

- Student Roll Number (used as the BST key)
- Name

- CGPA

Your program must:

- Allow insertion of new students into the BST
- Display the list of students in sorted order (ascending by roll number)
- Support search, deletion, and in-order traversal
- Be menu-driven and use recursion where appropriate

14. Simple City Connection Checker using Graph (DFS)

Topic: Graphs

Description:

You are tasked with simulating a simple city connection system using an unweighted, undirected graph.

Each city is a node, and a road between two cities is an edge.

Your goal is to allow users to:

- Add roads between cities
- Check if two cities are connected (directly or indirectly) using Depth-First Search (DFS)

REFERENCES:

1. Let Us C by Yashavant Kanetkar
2. Data Structures using c Aaron M.Tenenbaum, YedidyahLangsam, MosheJAugenstein.

Innovation and Entrepreneurship-I (Engineering Exploration)

Course Code: 25CC60
B.Tech II Year I Sem

L	T	P	C
0	0	2	1

Syllabus of II Year II Sem

COMPUTER ORIENTED STATISTICAL METHODS

(Common to CSE)

Course Code: 25HC13
B. Tech. II Year II Semester
Pre Requisites: Mathematics Knowledge at Pre-University Level

L	T	P	C
3	0	0	3

Course Objectives:

1. Concepts of probability, Probability distribution
2. Sampling distributions and inferential statistics.
3. The concept of Markov processes and equip students with the skills to analyze stochastic systems using transition probabilities and steady-state behavior.

Course Outcomes:

1. Apply the probability distributions to obtain mean and Standard deviation of a discrete random variable.
2. Make use of the continuous probability distributions to find the probability of a continuous random variable. Also Compute the Sampling distribution of a statistic
3. Estimate the mean of a population parameter and test the hypothesis concerning large samples.
4. Test of hypothesis concerning small samples and will be able to use chi-square test for goodness of fit.
5. Understand and apply the fundamentals of Markov chains, including transition probability matrices and steady-state analysis, to solve real-life stochastic problems.

UNIT-I: Random Variables and Discrete Probability Distributions: Concept of Discrete and Continuous Random Variables. Probability mass and density functions, expectation and variance, properties and problems. Binomial and Poisson Distributions.

UNIT-II: Continuous Probability Distributions: Gaussian (Normal) Distribution - properties and problems. Exponential Distribution. Fundamental Sampling Distributions: Random sampling, Sampling distribution of the mean (□ - known and unknown). Central limit theorem.

UNIT-III: Estimation: Point estimation and Interval estimation, single sample-estimating the mean for large samples, single sample – estimating a Proportion, Standard error of a point estimate, two samples-estimating ratio of two variances

Tests of Hypothesis for Large Samples: Tests of Hypothesis, Type–I and Type-II Errors, Hypothesis testing concerning one mean and two means and test of hypothesis concerning to one Proportion and difference of proportions.

UNIT-IV: Tests of Hypothesis for Small Samples: Student t-test, Hypothesis testing concerning one mean and two Means, F-test

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

TEXTBOOKS:

1. B. V. Ramana, Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11 th Reprint, 2010.
2. Miller and Freund's, Probability and Statistics for Engineers, 8th Edition, Pearson Educations
3. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, S Chand publications.

COMPUTER NETWORKS

Course Code: 25EC05

B.Tech II Year II Sem

Prerequisites: Programming for problem solving, Data Structures

L	T	P	C
3	0	0	3

Course Objectives:

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.
3. Elucidate the students about working and implementation of protocols at various layers in protocols stack.
4. Appreciating the protocol working by observing and analysing outputs of the packet sniffer.

Course Outcomes:

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.
5. Understanding working of the protocols through traces captured by a packet sniffer

UNIT - I

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: 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 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, 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 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

REFERENCES:

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

OPERATING SYSTEMS

Course Code: 25EC06
B.Tech II Year II Sem

L	T	P	C
3	0	0	3

Prerequisites:

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:

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

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

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

Interprocess 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.

REFERENCES:

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.

DESIGN AND ANALYSIS OF ALGORITHMS

Course Code: 25EC07
B.Tech II Year II Sem

L	T	P	C
3	0	0	3

Prerequisites: Programming for problem solving and Data Structures

Course Objectives:

1. Develop proficiency in evaluating algorithms using asymptotic notations, including best-, average-, and worst-case time/space complexities, and solving related recurrence relations.
2. Master various algorithmic strategies—divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound—identifying suitable use cases and demonstrating their application.
3. Critically assess and contrast different algorithms in terms of efficiency, scalability, and correctness through rigorous analytical reasoning and empirical evaluation.
4. Differentiate between tractable (polynomial-time) and intractable (super-polynomial or exponential-time) problems;
5. Identify and classify problems as P, NP, NP-hard, or NP-complete, and assess their relationships through polynomial-time reductions and Cook's theorem.

Course Outcomes:

1. Able to Apply space and time complexity analysis using asymptotic notations.
2. Able to Design divide-and-conquer algorithms and critically assess their runtime and space trade-offs.
3. Able to Device backtracking and dynamic programming solutions.
4. Able to Apply greedy methods and graph traversal algorithms
5. Able to Analyse and Design branch-and-bound algorithms for NP-hard problems

UNIT - I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation, and Little oh notation.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT - II

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue- Heaps, Heapsort

Backtracking: General method, applications, n-queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT - III

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

UNIT - IV

Greedy method: General method, applications- Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

UNIT - V

Branch and Bound: General method, applications - Travelling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni, and Rajasekaran, University Press.

REFERENCES:

1. Design and Analysis of algorithms, Aho, Ullman, and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C.Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and Sons.

Business Economics and Financial Analysis

Course Code: 25ZC01

B. Tech. II Year II Semester

L	T	P	C
3	0	0	3

Course Objectives:

1. To understand the basics of Business Economics at Micro level and Demand analysis in particular.
2. To understand the concept of production analysis, costs and Markets.
3. To understand the basic concepts of Accounting, Double entry system of Book keeping.
4. To understand the concepts of Capital expenditure, Revenue expenditure and Final accounts.
5. To enlighten students about the concept of Ratio Analysis.

Course Outcomes:

1. Explain the Basics of Economics and its relation to Business Economics (L1) -U1
2. Analyze the production function in terms of cost and revenue (L4)-U2
3. Outline the fundamentals of financial accounting and prepare financial statements (L5)- U3&amp;U4
4. Apply financial ratios to analyze the financial performance of the company (L3)-U5

UNIT I

INTRODUCTION TO BUSINESS ECONOMICS:

Definition, Nature and Scope of Business Economics–Demand Analysis: Demand Determinants, Law of Demand and its exceptions, Elasticity of Demand, Types of Elasticity of Demand and Demand Forecasting – Statistical and Non-Statistical techniques.

UNIT II

THEORY OF PRODUCTION COST ANALYSIS &amp; MARKETS:

Production Function – Isoquants and Isocosts, Internal and External Economies of Scale, Law of Returns Cost Analysis: Cost concepts, different types of costs, Break-even Analysis (BEA)- Determination of Break-Even Point (simple problems). Market structures: Types of competition, Features of Perfect competition, Monopoly and Monopolistic Competition, Pricing Methods and strategies.

UNIT III

FINANCIAL ACCOUNTING - I:

Accounting concepts and Conventions, Double-Entry system of Accounting, Accounting Cycle, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance.

UNIT IV

FINANCIAL ACCOUNTING – II:

Introduction to Final accounts, Revenue and Capital Expenditure, elements of Financial Statements, Preparation of Final Accounts with simple adjustments (simple problems).

UNIT-V

FINANCIAL ANALYSIS THROUGH RATIOS:

Concept of Ratio Analysis, Various Types of Ratios: Liquidity Ratios (short term solvency ratios), Leverage Ratios (long term solvency ratios), Turnover Ratios and Profitability Ratios (simple problems).

TEXTBOOKS:

1. A R Aryasri: Managerial Economics &amp; Financial Accounting, Tata Mc Graw Hill
2. A R Aryasri: Management Science, Tata Mc Graw Hill

REFERENCES:

1. S A Siddiqui &amp; A S Siddiqui, Managerial Economics &amp; Financial Analysis, New Age
2. Accountancy – I Tulasian Tata Mcgraw Hill Co
3. Koontz &amp;Weihrich: Essentials of Management, 6/e, TMH, 2005

COMPUTATIONAL MATHEMATICS LAB

(Using Python/MATLAB software)
(Common to all Branches)

Course Code: 25HC66
B. Tech. II Year II Semester
Pre Requisites: Matrices, Iterative methods and ordinary differential equations

L	T	P	C
0	0	2	1

Course Objectives:

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

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 and Higher order linear differential equations with constant coefficients

UNIT-I: Eigen values and Eigenvectors: 6P

Programs:

a) Finding real and complex Eigen values.
b) Finding Eigen vectors.

UNIT-II: Solution of Algebraic and Transcendental Equations 6P

Bisection method, Newton Raphson Method

Programs:

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

UNIT-III: Linear system of equations: 6P

Jacobi’s iteration method and Gauss-Seidal iteration method

Programs:

a) Solution of given system of linear equations using Jacobi’smethod
b) Solution of given system of linear equations using Gauss-Seidal method

UNIT-IV: First-Order ODEs 8P

Exact and non exact equations, Applications: exponential growth/decay, Newton&#39;s law of cooling.

Programs:

a) Solving exact and non-exact equations
b) Solving exponential growth/decay and Newton&#39;s law of cooling problems

UNIT-V: Higher order linear differential equations with constant coefficient 6P

Programs:

a) Solving homogeneous ODEs
b) Solving non homogeneous ODEs

TEXT BOOKS:

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

COMPUTER NETWORKS LAB

L	T	P	C
0	0	2	1

Course Code: 25EC64
B.Tech II Year II Sem

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:

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 Dijsktra'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**
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.

1. How to run Nmap scan
2. Operating System Detection using Nmap
3. Do the following using NS2 Simulator
    I. NS2 Simulator-Introduction
    II. Simulate to Find the Number of Packets Dropped
    III. Simulate to Find the Number of Packets Dropped by TCP/UDP
    IV. Simulate to Find the Number of Packets Dropped due to Congestion
    V. Simulate to Compare Data Rate & Throughput.
    VI. Simulate to Plot Congestion for Different Source/Destination

VII. 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

REFERENCES:

1. An Engineering Approach to Computer Networks, S.Keshav, 2nd Edition, Pearson Education Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

OPERATING SYSTEMS LAB

Course Code: 25EC65
B.Tech II Year II Sem

L	T	P	C
0	0	2	1

Prerequisites: Programming for Problem Solving, Computer Organization and Architecture, Operating Systems

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, interprocess communication and I/O in Unix.

Course Outcomes:

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, lseek, stat, fork, exit)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
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.

REFERENCES:

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.

REACT JS/ NODE JS/ EXPRESS JS/ MONGO DB

L	T	P	C
0	0	2	1

Course Code: 25FC10
B.Tech II Year II Sem
Prerequisites: Object Oriented Programming through Java, HTML Basics

Course Objectives:

1. To implement the static web pages using HTML and do client-side validation using JavaScript.
2. To introduce Node JS implementation for server-side programming.
3. To experiment with single page application development using React.
4. To deploy web applications to platforms like GitHub Pages and Netlify.
5. To automate continuous integration and deployment (CI/CD) pipelines using GitHub Actions.
6. To integrate NoSQL databases (MongoDB) with Node.js applications.

Course Outcomes:

1. Build a custom website using HTML, CSS, Bootstrap, and JavaScript.
2. Use advanced JavaScript features like ES6.
3. Create server-side applications using Node.js and Express.
4. Design and develop single-page applications using React.
5. Implement data persistence and management solutions for web applications using NoSQL databases.

Exercises:

1. Build a responsive web application for shopping cart with registration, login, catalog and cart pages using
CSS3 features, flex and grid.

2. Make the above web application a responsive web application using Bootstrap framework.

3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1.

4. Explore the features of ES6 like arrow functions, callbacks, promises, async/await. Implement an
application for reading the weather information from openweathermap.org and display the information in
the form of a graph on the web page.

5. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.

6. Develop an express web application that can interact with REST API to perform CRUD operations on
student data. (Use Postman)

7. For the above application create authorized end points using JWT (JSON Web Token).

8. Create a react application for the student management system having registration, login, contact, about
pages and implement routing to navigate through these pages.

9. Create a service in react that fetches the weather information from openweathermap.org and the display
the current and historical weather information using graphical representation using chart.js

10. Create a TODO application in react with necessary components and link it into GitHub and deploy on
Netlify or similar service.

11. Implement unit testing in React (Jest) and Node (Mocha). Ensure coverage of key modules.

12. Automate deployment pipeline using GitHub Actions for any one of the above applications.

13. Convert a React application to TypeScript and demonstrate basic type usage.

14. Integrate MongoDB with Node.js backend using Mongoose to perform full CRUD on collections.

REFERENCES/ WEBSITES:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.
2. https://web.dev/learn/html
3. https://web.dev/learn/css
4. https://getbootstrap.com/
5. https://nodejs.org/en/docs/
6. https://reactjs.org/
7. https://expressjs.com/
8. https://www.mongodb.com/docs/manual/

Innovation and Entrepreneurship-II (Prototype Realization)

Course Code: 25CC61
B.Tech II Year II Sem

L	T	P	C
0	0	2	1