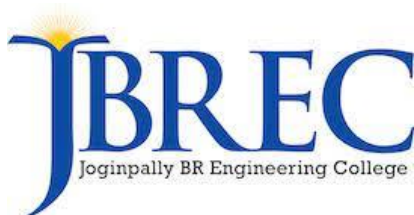


JOGINPALLY B.R. ENGINEERING COLLEGE
(UGC AUTONOMOUS)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

R23 B.Tech. SYLLABUS

VISION AND MISSION OF THE INSTITUTION

VISION

To be a global leader in education and research in Engineering and Management.

MISSION:

M1: To impart high quality professional education to mould the learners into globally competitive engineers/managers who are professionally deft, intellectually adept and socially responsible.

M2: To collaborate with industries and research organizations and excel in the emerging areas of research.

VISION AND MISSION OF THE DEPARTMENT

VISION:

To impart quality education for moulding students as competent professionals with knowledge of evolving technologies to meet the demands of industry and society.

MISSION:

M1: To impart latest knowledge for successful careers in Computer Science and Engineering.

M2: To undertake interdisciplinary research to meet eternally changing global industry needs.

M3: To inculcate lifelong learning, team work and ethics for the overall benefit of the society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- PEO1:** To apply the knowledge of computer science and engineering to solve real world problems by using various software tools and technologies.
- PEO2:** To develop a zest for research in essential technologies of Computer Science and Engineering.
- PEO3:** To work for a successful career in industry and serve its various needs across multiple disciplines.
- PEO4:** To contribute effectively to societal needs with due regard to ethics and values.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- PSO1:** Analyze and develop computer programs in areas related to system software, multimedia, web design, big data analytics and networking for efficient design of products of varying complexity.
- PSO2:** Apply strategies and standard practices in software project development using latest programming environments to deliver quality products for business success.
- PSO3:** Employ modern computer languages, environments and platforms for creating innovative career paths for entrepreneurship and higher studies.

PROGRAM OUTCOMES (POs)

- PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- PO2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with

appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

- PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
- PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

JOGINPALLY B.R. ENGINEERING COLLEGE, HYDERABAD
(UGC AUTONOMOUS)

B.Tech. in COMPUTER SCIENCE AND ENGINEERING
COURSE STRUCTURE & SYLLABUS (R23 Regulations)

Applicable from AY 2023-24 Batch

I Year I Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA 101 BS	Matrices and Calculus	3	1	0	4
2.	CH 102 BS	Engineering Chemistry	3	1	0	4
3.	CS 103 ES	Programming for Problem Solving	3	0	0	3
4.	EE 104 ES	Basic Electrical Engineering	2	0	0	2
5.	ME 105 ES	Computer Aided Engineering Graphics	1	0	4	3
6.	CS 106 ES	Elements of Computer Science & Engineering	0	0	2	1
7.	CH 107 BS	Engineering Chemistry Laboratory	0	0	2	1
8.	CS 108 ES	Programming for Problem Solving Laboratory	0	0	2	1
9.	EE 109 ES	Basic Electrical Engineering Laboratory	0	0	2	1
		Induction Program				
		Total	12	2	12	20

I Year II Semester

S. No.	Course Code	Course	L	T	P	Credits
1.	MA 201 BS	Ordinary Differential Equations and Vector Calculus	3	1	0	4
2.	PH 202 BS	Applied Physics	3	1	0	4
3.	ME 203 ES	Engineering Workshop	0	1	3	2.5
4.	EN 204 HS	English for Skill Enhancement	2	0	0	2
5.	EC 205 ES	Electronic Devices and Circuits	2	0	0	2
6.	CS 206 ES	Python Programming Laboratory	0	1	2	2
7.	PH 207 BS	Applied Physics Laboratory	0	0	3	1.5
8.	EN 208 HS	English Language and Communication Skills Laboratory	0	0	2	1
9.	CS 209 ES	IT Workshop	0	0	2	1
10.	*MC210	Environmental Science	3	0	0	0
		Total	13	4	12	20

II YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	EC 301 PC	Digital Electronics	3	0	0	3
2	CS 302 PC	Data Structures	3	0	0	3
3	CS 303 PC	Computer Oriented Statistical Methods	3	1	0	4
4	CS 304 PC	Computer Organization and Architecture	3	0	0	3
5	CS 305 PC	Object Oriented Programming through Java	3	0	0	3
6	CS 306 PC	Data Structures Lab	0	0	3	1.5
7	CS 307 PC	Object Oriented Programming through Java Lab	0	0	3	1.5
8	CS 308 PC	Data visualization- R Programming/ Power BI	0	0	2	1
9	*MC309	Gender Sensitization Lab	0	0	2	0
		Total	15	1	10	20

II YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS 401 ES	Discrete Mathematics	3	0	0	3
2	MB 402 HS	Business Economics & Financial Analysis	3	0	0	3
3	CS 403 PC	Operating Systems	3	0	0	3
4	CS 404 PC	Database Management Systems	3	0	0	3
5	CS 405 PC	Software Engineering	3	0	0	3
6	CS 406 PC	Operating Systems Lab	0	0	2	1
7	CS 407 PC	Database Management Systems Lab	0	0	2	1
8	CS 408 PW	Real-time Research Project/ Societal Related Project	0	0	4	2
9	CS 409 PC	Node JS/ React JS/ Django	0	0	2	1
10	*MC410	Constitution of India	3	0	0	0
Total			18	0	10	20

III YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS 501 PC	Design and Analysis of Algorithms	3	1	0	4
2	CS 502 PC	Computer Networks	3	0	0	3
3	CS 503 PC	DevOps	3	0	0	3
4		Professional Elective-I	3	0	0	3
5		Professional Elective -II	3	0	0	3
6	CS 504 PC	Computer Networks Lab	0	0	2	1
7	CS 505 PC	DevOps Lab	0	0	2	1
8	EG 506 HS	Advanced English Communication Skills Lab	0	0	2	1
9	CS 507 PC	UI design- Flutter	0	0	2	1
10	*MC508	Intellectual Property Rights	3	0	0	0
Total			18	1	8	20

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS 601 PC	Machine Learning	3	0	0	3
2	CS 602 PC	Formal Languages and Automata Theory	3	0	0	3
3	CS 603 PC	Artificial Intelligence	3	0	0	3
4		Professional Elective – III	3	0	0	3
5		Open Elective-I	3	0	0	3
6	CS 604 PC	Machine Learning Lab	0	0	2	1
7	CS 605 PC	Artificial Intelligence Laboratory	0	0	2	1
8		Professional Elective-III Lab	0	0	2	1
9	CS 606 PW	Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	*MC607	Environmental Science	3	0	0	0
Total			18	0	8	20

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

IV YEAR I SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	CS 701 PC	Cryptography and Network Security	3	0	0	3
2	CS 702 PC	Compiler Design	3	0	0	3
3		Professional Elective -IV	3	0	0	3
4		Professional Elective -V	3	0	0	3
5		Open Elective - II	3	0	0	3
6	CS 703 PC	Cryptography and Network Security Lab	0	0	2	1
7	CS 704 PC	Compiler Design Lab	0	0	2	1
8	CS 705 PW	Project Stage - I	0	0	6	3
		Total Credits	15	0	10	20

IV YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	MB 801 HS	Organizational Behavior	3	0	0	3
2		Professional Elective – VI	3	0	0	3
3		Open Elective – III	3	0	0	3
4	CS 802 PW	Project Stage – II including Seminar	0	0	22	11
		Total Credits	9	0	22	20

***MC – Satisfactory/Unsatisfactory**

#Skill Course - 1 credit with 2 Practical Hours

Professional Elective - I

CS 509 PE	Quantum Computing
CS 510 PE	Advanced Computer Architecture
CS 511 PE	Data Analytics
CS 512 PE	Image Processing
CS 513 PE	Principles of Programming Languages

Professional Elective - II

CS 514 PE	Computer Graphics
CS 515 PE	Embedded Systems
CS 516 PE	Information Retrieval Systems
CS 517 PE	Distributed Databases
CS 518 PE	Natural Language Processing

Professional Elective - III

CS 608 PE	Full Stack Development
CS 609 PE	Internet of Things
CS 610 PE	Scripting Languages
CS 611 PE	Mobile Application Development
CS 612 PE	Software Testing Methodologies

Courses in PE - III and PE - III Lab must be in 1-1 correspondence.

Professional Elective -IV

CS 706 PE	Graph Theory
CS 707 PE	Cyber Security
CS 708 PE	Soft Computing
CS 709 PE	Cloud Computing
CS 710 PE	Ad hoc & Sensor Networks

Professional Elective -V

CS 711 PE	Advanced Algorithms
CS 712 PE	Agile Methodology
CS 713 PE	Robotic Process Automation
CS 714 PE	Blockchain Technology
CS 715 PE	Software Process & Project Management

Professional Elective – VI

CS 803 PE	Computational Complexity
CS 804 PE	Distributed Systems
CS 805 PE	Deep Learning
CS 806 PE	Human Computer Interaction
CS 807 PE	Cyber Forensics

Open Elective -1

CS 618 OE	Data Structures
CS 619 OE	Database Management Systems

Open Elective -2:

CS 716 OE	Operating Systems
CS 717 OE	Software Engineering

Open Elective -3:

CS 808 OE	Algorithms Design and Analysis
CS 809 OE	Introduction to Computer Networks

MA 101BS: MATRICES AND CALCULUS**B.Tech. I Year I Sem.**

L	T	P	C
3	1	0	4

Pre-requisites: Mathematical Knowledge at pre-university level**Course Objectives:**

- Represent and analyze linear systems using matrix algebra and test their consistency.
- Transform quadratic forms into canonical form through orthogonal transformations for simplification and analysis.
- Apply the Mean Value Theorems and Taylor's Series to approximate and analyze functions. Evaluate complex integrals using special functions like Beta and Gamma functions.
- Determine local extrema of functions involving two or more variables using optimization techniques.
- Understand and differentiate between double and triple integrals in Cartesian and polar coordinate systems to solve area and volume problems.

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

- Write the matrix form of linear system of equations and test the consistency of the given system
- Reduce the quadratic form to canonical form using orthogonal transformations.
- Solve the applications on the mean value theorems and expand Taylor's series. Evaluate the improper integrals using Beta and Gamma functions
- Find the Maxima and Minima of functions of two variables and three variables.
- Compare the difference between double integrals and triple integrals in cartesian and polar Coordinates.

UNIT - I: Matrices**10 L**

Rank of a matrix by Echelon form and Normal form, Inverse of Non-singular matrices by Gauss-Jordan method, System of linear equations: Solving system of Homogeneous and Non-Homogeneous equations by Gauss elimination method, Gauss Seidel Iteration Method.

UNIT - II: Eigen values and Eigen vectors**10 L**

Linear Transformation and Orthogonal Transformation: Eigenvalues, Eigenvectors and their properties, Diagonalization of a matrix, Cayley-Hamilton Theorem (without proof), finding inverse and power of a matrix by Cayley-Hamilton Theorem, Quadratic forms and Nature of the Quadratic Forms, Reduction of Quadratic form to canonical forms by Orthogonal Transformation.

UNIT - III: Calculus**10 L**

Mean value theorems: Rolle's theorem, Lagrange's Mean value theorem with their Geometrical Interpretation and applications, Cauchy's Mean value Theorem, Taylor's Series.

Applications of definite integrals to evaluate surface areas and volumes of revolutions of curves (Only in Cartesian coordinates), Definition of Improper Integral: Beta and Gamma functions and their applications.

UNIT - IV: Multivariable Calculus (Partial Differentiation and applications)**10 L**

Definitions of Limit and continuity.

Partial Differentiation: Euler's Theorem, Total derivative, Jacobian, Functional dependence & independence.

Applications: Maxima and minima of functions of two variables and three variables using method of Lagrange multipliers.

UNIT-V: Multivariable Calculus (Integration)**8 L**

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

Applications: Areas (by double integrals) and volumes (by double integrals and triple integrals).

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CH 102 BS : ENGINEERING CHEMISTRY**B.Tech. I Year I Sem.****L T P C****3 1 0 4**

Course Objectives: This course is designed to provide students with a deep understanding of fundamental physical principles and their applications in science and engineering. It aims to:

- Develop a conceptual understanding of quantum mechanics and solid-state physics to differentiate between conductors, semiconductors, and insulators based on their electronic properties.
- Enable students to recognize the significance and working principles of semiconductor devices in various technological applications.
- Introduce the essential characteristics and practical uses of dielectric, magnetic, and energy materials in advanced engineering systems.
- Foster an appreciation for the structure, synthesis, and applications of nanomaterials in modern science and technology.
- Impart knowledge on the working principles and uses of lasers and optical fibers, emphasizing their significance in communication, medicine, and industrial applications.

Course Outcomes:

- Understand the physical world from the fundamental point of view by the concepts of Quantum mechanics and visualize the difference between a conductor, a semiconductor, and an insulator by classification of solids.
- Identify the role of semiconductor devices in science and engineering applications.
- Explore the fundamental properties of dielectric, magnetic materials, and energy materials for their applications.
- Appreciate the features and applications of Nanomaterials
- Understand various aspects of Lasers and Optical fiber and their applications in diverse fields

UNIT - I: Water and its treatment: [8]

Introduction to hardness of water – Estimation of hardness of water by complexometric method and related numerical problems. Potable water and its specifications - Steps involved in the treatment of potable water - Disinfection of potable water by chlorination and break - point chlorination. Defluoridation - Determination of F⁻ ion by ion- selective electrode method.

Boiler troubles: Sludges, Scales 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 water – Reverse osmosis.

UNIT – II Battery Chemistry & Corrosion [8]

Introduction - Classification of batteries- primary, secondary and reserve batteries with examples. Basic requirements for commercial batteries. Construction, working and applications of: Zn-air and Lithium ion battery, Applications of Li-ion battery to electrical vehicles. Fuel Cells- Differences between battery and a fuel cell, Construction and applications of Methanol Oxygen fuel cell and Solid oxide fuel cell. Solar cells - Introduction and applications of Solar cells.

Corrosion: Causes and effects of corrosion – theories of chemical and electrochemical corrosion – mechanism of electrochemical corrosion, Types of corrosion: Galvanic, water-line and pitting corrosion. Factors affecting rate of corrosion, Corrosion control methods- Cathodic protection – Sacrificial anode and impressed current methods.

UNIT - III: Polymeric materials: [8]

Definition – Classification of polymers with examples – Types of polymerization – addition (free radical addition) and condensation polymerization with examples – Nylon 6:6, Terylene **Plastics:** Definition and characteristics- thermoplastic and thermosetting plastics, Preparation, Properties and engineering applications of PVC and Bakelite, Teflon, Fiber reinforced plastics (FRP). **Rubbers:** Natural rubber and its vulcanization.

Elastomers: Characteristics –preparation – properties and applications of Buna-S, Butyl and Thiokol rubber.

Conducting polymers: Characteristics and Classification with examples-mechanism of conduction in trans-polyacetylene and applications of conducting polymers.

Biodegradable polymers: Concept and advantages - Polylactic acid and poly vinyl alcohol and their applications.

UNIT - IV: Energy Sources: [8]

Introduction, Calorific value of fuel – HCV, LCV- Dulong's formula. Classification- solid fuels: coal – analysis of coal – proximate and ultimate analysis and their significance. Liquid fuels – petroleum and its refining, cracking types – moving bed catalytic cracking. Knocking – octane and cetane rating, synthetic petrol - Fischer-Tropsch's process; Gaseous fuels – composition and uses of natural gas, LPG and CNG, Biodiesel – Transesterification, advantages.

UNIT - V: Engineering Materials: [8]

Cement: Portland cement, its composition, setting and hardening.

Smart materials and their engineering applications

Shape memory materials- Poly L- Lactic acid. Thermoresponsive materials- Polyacryl amides, Poly vinyl amides

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

TEXT BOOKS:

1. Engineering Chemistry by P.C. Jain and M. Jain, Dhanpatrai Publishing Company, 2010
2. Engineering Chemistry by Rama Devi, Venkata Ramana Reddy and Rath, Cengage learning, 2016
3. A text book of Engineering Chemistry by M. Thirumala Chary, E. Laxminarayana and K. Shashikala, Pearson Publications, 2021.
4. Textbook of Engineering Chemistry by Jaya Shree Anireddy, Wiley Publications.

REFERENCE BOOKS:

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

CS 103 ES : PROGRAMMING FOR PROBLEM SOLVING**B.Tech. I Year I Sem.****L T P C****3 0 0 3****Course Objectives:**

- To learn the fundamentals of computers and various steps in program development.
- Develop proficiency in using basic data types such as arrays, pointers, strings, and structures to write efficient C programs.
- Enhance the understanding and application of preprocessor directives and file handling to promote modular, readable, and persistent programs.
- Encourage decomposition of complex problems into functions to create modular and reusable code for better maintainability.
- Build a strong foundation in implementing fundamental searching and sorting algorithms, enabling efficient data processing and manipulation.

Course Outcomes: The student will learn

- To write algorithms and to draw flowcharts for solving problems.
- To use arrays, pointers, strings and structures to write C programs
- Understand and effectively utilize preprocessor directives, file handling techniques in C to enhance code modularity, readability, and data management
- To decompose a problem into functions and to develop modular reusable code.
- To understand and develop Searching and sorting problems.

UNIT - I: 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, Runtime Errors in execution, object and executable code, Operators, expressions and precedence, Expression evaluation, Storage classes (auto, extern, static and register), type conversion, The main method and command line arguments Bitwise operations: Bitwise AND, OR, XOR and NOT operators, Type conversion

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, break, continue, goto, Iteration with for, while, do- while loops

I/O: Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr. Command line arguments

UNIT - II: Arrays, Strings, Structures and Pointers:

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

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

UNIT - III: Preprocessor and File handling in C:

Preprocessor: Commonly used Preprocessor commands like include, define, undef, if, ifdef, ifndef 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.

UNIT - IV: Function and Dynamic Memory Allocation:

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
Dynamic memory allocation: Allocating and freeing memory, Allocating memory for arrays of different data types

UNIT - V: Searching and Sorting:

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

EE 104 ES: BASIC ELECTRICAL ENGINEERING**B.Tech. I Year I Sem.**

L	T	P	C
2	0	0	2

Prerequisites: Mathematics**Course Objectives:** It is designed to:

- Provide foundational knowledge to analyze and solve electrical circuits using basic laws and theorems.
- Enable students to understand the behavior of electric and magnetic circuits and their practical implications.
- Develop the ability to analyze and evaluate the performance of single-phase and three-phase transformers, considering efficiency, losses, and equivalent circuit models.
- Impart fundamental understanding of electrical machines such as DC machines, induction motors, and synchronous machines.
- Familiarize students with low-voltage electrical installation systems, components, and safety measures for residential and industrial applications.

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

- Analyze and solve electrical circuits and theorems
- Understand and analyze basic Electric and Magnetic circuits
- Analyze the performance characteristics of transformers, including losses, efficiency, and equivalent circuit analysis.
- Understand the working principles of Electrical Machines
- Apply knowledge of low voltage electrical installation components

UNIT-I:

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

UNIT-II:

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

UNIT-III:

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

UNIT-IV:

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

UNIT-V:

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

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

ME 105 ES: COMPUTER AIDED ENGINEERING GRAPHICS**B.Tech. I Year I Sem.**

L	T	P	C
1	0	4	3

Course Objectives:

- Introduce students to the use of computer-aided drafting (CAD) tools for creating accurate 2D and 3D representations of engineering objects.
- Enable students to sketch geometric shapes, including conics and various types of solids, essential for technical communication.
- Emphasize the importance and application of sectional views and the development of surfaces for understanding internal features and fabrication processes.
- Train students to read, interpret, and analyze standard engineering drawings as a means of technical communication.
- Develop the ability to convert orthographic views into isometric projections and vice versa, both manually and through CAD tools, enhancing spatial visualization and design interpretation skills.

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

- Apply computer aided drafting tools to create 2D and 3D objects
- sketch conics and different types of solids
- Appreciate the need of Sectional views of solids and Development of surfaces of solids
- Read and interpret engineering drawings
- Conversion of orthographic projection into isometric view and vice versa manually and by using computer aided drafting

UNIT – I:

Introduction to Engineering Graphics: Principles of Engineering Graphics and their Significance, Scales – Plain & Diagonal, Conic Sections including the Rectangular Hyperbola – General method only. Cycloid, Epicycloid and Hypocycloid, Introduction to Computer aided drafting – views, commands and conics

UNIT- II:

Orthographic Projections: Principles of Orthographic Projections – Conventions – Projections of Points and Lines, Projections of Plane regular geometric figures. Auxiliary Planes. Computer aided orthographic projections – points, lines and planes

UNIT – III:

Projections of Regular Solids – Auxiliary Views - Sections or Sectional views of Right Regular Solids – Prism, Cylinder, Pyramid, Cone – Auxiliary views, Computer aided projections of solids – sectional views

UNIT – IV:

Development of Surfaces of Right Regular Solids – Prism, Cylinder, Pyramid and Cone, Development of surfaces using computer aided drafting

UNIT – V:

Isometric Projections: Principles of Isometric Projection – Isometric Scale – Isometric Views – Conventions – Isometric Views of Lines, Plane Figures, Simple and Compound Solids – Isometric Projection of objects having non- isometric lines. Isometric Projection of Spherical Parts. Conversion of Isometric Views to Orthographic Views and Vice-versa –Conventions. Conversion of orthographic projection into isometric view using computer aided drafting.

TEXT BOOKS:

1. Engineering Drawing N.D. Bhatt / Charotar
2. Engineering Drawing and graphics Using AutoCAD Third Edition, T. Jeyapoovan, Vikas: S. Chand and company Ltd.

REFERENCE BOOKS:

1. Engineering Drawing, Basant Agrawal and C M Agrawal, Third Edition McGraw Hill
2. Engineering Graphics and Design, WILEY, Edition 2020
3. Engineering Drawing, M. B. Shah, B.C. Rane / Pearson.
4. Engineering Drawing, N. S. Parthasarathy and Vela Murali, Oxford

5. Computer Aided Engineering Drawing – K Balaveera Reddy et al – CBS Publishers

Note: - External examination is conducted in conventional mode and internal evaluation to be done by both conventional as well as using computer aided drafting.

CS 106 ES: ELEMENTS OF COMPUTER SCIENCE AND ENGINEERING**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

Course Objective: To provide an overview of the subjects of computer science and engineering.

- Introduce students to the basic functional units and working principles of a computer system, enabling a thorough understanding of its architecture.
- Develop problem-solving skills through the fundamentals of programming, data structures, and algorithmic thinking.
- Provide an overview of operating systems and database systems, emphasizing their roles, types, and applications in computing.
- Create awareness about computer networks, the Internet, World Wide Web (WWW), and cyber security principles to ensure safe and responsible use of digital resources.
- Familiarize students with emerging technologies, including autonomous systems and artificial intelligence, highlighting their applications and relevance in modern engineering and societal contexts.

Course Outcomes:

- Know the working principles of functional units of a basic Computer
- Understand program development, the use of data structures and algorithms in problem solving.
- Know the need and types of operating system, database systems.
- Understand the significance of networks, internet, WWW and cyber security.
- Understand Autonomous systems, the application of artificial intelligence.

UNIT – I

Basics of a Computer – Hardware, Software, Generations of computers. Hardware - functional units, Components of CPU, Memory – hierarchy, types of memory, Input and output devices. Software – systems software, application software, packages, frameworks, IDEs.

UNIT – II

Software development – waterfall model, Agile, Types of computer languages – Programming, markup, scripting Program Development – steps in program development, flowcharts, algorithms, data structures – definition, types of data structures

UNIT – III

Operating systems: Functions of operating systems, types of operating systems, Device & Resource management

Database Management Systems: Data models, RDBMS, SQL, Database Transactions, data centers, cloud services

UNIT – IV

Computer Networks: Advantages of computer networks, LAN, WAN, MAN, internet, WiFi, sensor networks, vehicular networks, 5G communication.

World Wide Web – Basics, role of HTML, CSS, XML, Tools for web designing, Social media, Online social networks.

Security – information security, cyber security, cyber laws

UNIT – V

Autonomous Systems: IoT, Robotics, Drones, Artificial Intelligence – Learning, Game Development, natural language processing, image and video processing.

Cloud Basics

TEXT BOOK:

1. Invitation to Computer Science, G. Michael Schneider, Macalester College, Judith L. Gersting University of Hawaii, Hilo, Contributing author: Keith Miller University of Illinois, Springfield.

REFERENCE BOOKS:

1. Fundamentals of Computers, Reema Thareja, Oxford Higher Education, Oxford University Press.
2. Introduction to computers, Peter Norton, 8th Edition, Tata McGraw Hill.
3. Computer Fundamentals, Anita Goel, Pearson Education India, 2010.
4. Elements of computer science, Cengage.

CH 107 BS: ENGINEERING CHEMISTRY LABORATORY**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

Course Objectives: The course consists of experiments related to the principles of chemistry required for engineering student. The student will learn:

- Estimation of hardness of water to check its suitability for drinking purpose.
- Students are able to perform estimations of acids and bases using conductometry, potentiometry and pH metry methods.
- Students will learn to prepare polymers such as Bakelite and nylon-6 in the laboratory.
- Students will learn the exploration of advanced and virtual experiments by modeling modern technologies such as fuel cells, smart materials in biomedical applications, batteries for electric vehicles, and solar cell operations with their real-world applications.

Course Outcomes: The experiments will make the student gain skills on:

- Determination of parameters like hardness of water and rate of corrosion of mild steel in various conditions.
- Able to perform methods such as conductometry, potentiometry and pH metry in order to find out the concentrations or equivalence points of acids and bases.
- Students are able to prepare polymers like bakelite and nylon-6.
- Estimations saponification value, surface tension and viscosity of lubricant oils.
- Students are able to prepare models of virtual experiments Fuel cell and its working, Smart materials for Biomedical applications, Batteries for electrical vehicles and Functioning of solar cell and its applications

List of Experiments:

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

II. Conductometry: Estimation of the concentration of an acid by Conductometry.

III. Potentiometry: Estimation of the amount of Fe^{+2} by Potentiometry.

IV. pH Metry: Determination of an acid concentration using pH meter.

V. Preparations:

1. Preparation of Bakelite.
2. Preparation Nylon – 6.

VI. Lubricants:

1. Estimation of acid value of given lubricant oil.
2. Estimation of Viscosity of lubricant oil using Ostwald's Viscometer.

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 its working.
2. Smart materials for Biomedical applications
3. Batteries for electrical vehicles.
4. Functioning of solar cell and its applications.

REFERENCE BOOKS:

1. Lab manual for Engineering chemistry by B. Ramadevi and P. Aparna, S Chand Publications, New Delhi (2022)
2. Vogel's text book of practical organic chemistry 5th edition
3. Inorganic Quantitative analysis by A.I. Vogel, ELBS Publications.
4. College Practical Chemistry by V.K. Ahluwalia, Narosa Publications Ltd. New Delhi (2007).

CS 108 ES: PROGRAMMING FOR PROBLEM SOLVING LABORATORY**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: The students will learn the following:

- Train students to design step-by-step algorithms for simple computational problems and translate them into syntactically and logically correct C programs.
- Enable students to identify, debug, and correct syntax and logical errors through compiler feedback and program testing.
- Provide practical experience in representing and manipulating data using arrays, strings, structures, and pointers of various types.
- Introduce file handling concepts by allowing students to create, read, and write data using simple text and binary files.
- Promote modular and structured programming by encouraging the use of user-defined functions to improve code reusability and clarity.

Course Outcomes: The candidate is expected to be able to:

- Formulate the algorithms for simple problems and translate it to a working and correct program
- Identify and correct logical errors encountered during execution and correct syntax errors as reported by the compilers
- Represent and manipulate data with arrays, strings, structures and use pointers of different types
- Create, read and write to and from simple text and binary files
- Able to modularize the code with functions so that they can be reused

Practice sessions:

- 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.
- Write a simple program that converts one given data type to another using auto conversion and casting. Take the values from standard input.
- Write a program to print ASCII value of a character.

Simple numeric problems:

- Write a program for finding the max and min from the three numbers.
- Write the program for the simple, compound interest.
- 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.
- 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
```
- Write a program that shows the binary equivalent of a given positive number between 0 to 255.

Expression Evaluation:

- a. 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 ($= 9.8 m/s^2$)).
- b. 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)
- c. Write a program that finds if a given number is a prime number
- d. Write a C program to find the sum of individual digits of a positive integer and test given number is palindrome.
- e. 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.
- f. Write a C program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
- g. Write a C program to find the roots of a Quadratic equation.
- h. Write a C program to calculate the following, where x is a fractional value. i. $1 - x/2 + x^2/4 - x^3/6$
- j. Write a C program to read in two numbers, x and n , and then compute the sum of this geometric progression: $1 + x + x^2 + x^3 + \dots + x^n$. For example: if n is 3 and x is 5, then the program computes $1 + 5 + 25 + 125$.

Arrays, Pointers and Functions:

- 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:
- d. Addition of Two Matrices
- e. Multiplication of Two Matrices
- f. Transpose of a matrix with memory dynamically allocated for the new matrix as row and column counts may not be the same.
- g. Write C programs that use both recursive and non-recursive functions
- h. To find the factorial of a given integer.
- i. To find the GCD (greatest common divisor) of two given integers.
- j. To find x^n
- k. Write a program for reading elements using a pointer into an array and display the values using the array.
- l. Write a program for display values reverse order from an array using a pointer.
- m. Write a program through a pointer variable to sum of n elements from an array.

Files:

- 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.
- d. Write a C program that does the following:
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)
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)
The program should then read all 10 values and print them back.

- e. 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).

Strings:

- a. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
- b. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent
- c. Write a C program that uses functions to perform the following operations:
- d. To insert a sub-string into a given main string from a given position.
- e. To delete n Characters from a given position in a given string.
- f. 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.)
- g. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn't contain ch.
- h. Write a C program to count the lines, words and characters in a given text.

Miscellaneous:

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

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

Sorting and Searching:

- a. Write a C program that uses non recursive function to search for a Key value in a given
- b. list of integers using linear search method.
- c. Write a C program that uses non recursive function to search for a Key value in a given
- d. sorted list of integers using binary search method.
- e. Write a C program that implements the Bubble sort method to sort a given list of
- f. integers in ascending order.
- g. Write a C program that sorts the given array of integers using selection sort in descending order
- h. Write a C program that sorts the given array of integers using insertion sort in ascending order
- i. Write a C program that sorts a given array of names

TEXT BOOKS:

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

REFERENCE BOOKS:

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

EE 109 ES: BASIC ELECTRICAL ENGINEERING LABORATORY**B.Tech. I Year I Sem.**

L	T	P	C
0	0	2	1

Prerequisites: Basic Electrical Engineering**Course Objectives:**

- Enable students to apply Kirchhoff's Voltage and Current Laws (KVL, KCL) and basic network theorems to analyze and verify the operation of electrical circuits.
- Facilitate understanding of both transient and steady-state responses of RL, RC, and RLC circuits under DC and AC excitations through experimental observations.
- Develop the ability to evaluate key performance parameters of single-phase transformers, including voltage, current, power, efficiency, and voltage regulation.
- Provide hands-on experience in analyzing the operational characteristics of DC and AC machines, including speed-torque behavior.
- Equip students with skills to measure active and reactive power in three-phase circuits and understand transformer connections in three-phase electrical systems.

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

- Apply KVL, KCL, and network theorems to analyze and verify the behavior of electrical circuits.
- Analyze the transient and steady-state response of RL, RC, and RLC circuits under DC and AC excitation.
- Evaluate the performance of single-phase transformers through voltage, current, power, efficiency, and regulation.
- Analyze the characteristics of DC and AC machines including speed-torque behavior.
- Measure active/reactive power in three-phase circuits and analyze transformer connections using three-phase systems.

List of experiments/demonstrations:**PART- A (compulsory)**

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

PART-B (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.

MA 201 BS: ORDINARY DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS**B.Tech. I Year II Sem.**

L	T	P	C
3	1	0	4

Pre-requisites: Mathematical Knowledge at pre-university level**Course Objectives:** To learn

- Develop the ability to identify and solve exact and non-exact first-order differential equations.
- Enable students to solve higher-order differential equations and apply them to real-world engineering and physical problems.
- Introduce Laplace transform techniques to solve ordinary differential equations (ODEs) efficiently and handle initial value problems.
- Provide conceptual understanding and analytical tools to study scalar and vector fields using differential operators such as gradient, divergence, and curl.
- Train students to evaluate line, surface, and volume integrals and transform them using fundamental theorems of vector calculus like Green's, Stokes', and Gauss' theorems.

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

- Identify whether the given differential equation of first order is exact or not
- Solve higher differential equation and apply the concept of differential equation to real world problems.
- Use the Laplace transforms techniques for solving ODE's.
- Analyze and understand the behavior of vector and scalar fields using differential operators such as gradient, divergence and curl.
- Evaluate the line, surface and volume integrals and converting them from one to another

UNIT-I: First Order ODE**8 L**

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

UNIT-II: Ordinary Differential Equations of Higher Order**10 L**

Second order linear differential equations with constant coefficients: Non-Homogeneous terms of the type e^{ax} , $\sin ax$, $\cos ax$, polynomials in x , $e^{ax}V(x)$ and $x V(x)$, method of variation of parameters, Equations reducible to linear ODE with constant coefficients: Legendre's equation, Cauchy-Euler equation. Applications: Electric Circuits

UNIT-III: Laplace transforms**10 L**

Laplace Transforms: Laplace Transform of standard functions, First shifting theorem, Second shifting theorem, Unit step function, Dirac delta function, Laplace transforms of functions when they are multiplied and divided by 't', Laplace transforms of derivatives and integrals of function, Evaluation of integrals by Laplace transforms, Laplace transform of periodic functions, Inverse Laplace transform by different methods, convolution theorem (without proof). Applications: solving Initial value problems by Laplace Transform method.

UNIT-IV: Vector Differentiation**10 L**

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

UNIT-V: Vector Integration**10 L**

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

PH 202 BS: APPLIED PHYSICS**B.Tech. I Year II Sem.**

L	T	P	C
3	1	0	4

Pre-requisites: 10 + 2 Physics**Course Objectives:** The objectives of this course for the student are to:

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

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

- Understand physical world from fundamental point of view by the concepts of Quantum mechanics and visualize the difference between conductor, semiconductor, and an insulator by classification of solids.
- Identify the role of semiconductor devices in science and engineering Applications.
- Explore the fundamental properties of dielectric, magnetic materials and energy for their applications.
- Appreciate the features and applications of Nanomaterials.
- Understand various aspects of Lasers and Optical fiber and their applications in diverse fields.

UNIT - I: QUANTUM PHYSICS AND SOLIDS

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

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

UNIT - II: SEMICONDUCTORS AND DEVICES

Intrinsic and extrinsic semiconductors – Hall effect - direct and indirect band gap semiconductors - construction, principle of operation and characteristics of P-N Junction diode, Zener diode and bipolar junction transistor (BJT)–LED, PIN diode, avalanche photo diode (APD) and solar cells, their structure, materials, working principle and characteristics.

UNIT - III: DIELECTRIC, MAGNETIC AND ENERGY MATERIALS

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

Magnetic Materials: Hysteresis - soft and hard magnetic materials - magnetostriction, magnetoresistance - applications - bubble memory devices, magnetic field sensors and multiferroics. Energy Materials: Conductivity of liquid and solid electrolytes- superionic conductors - materials and electrolytes for super capacitors - rechargeable ion batteries, solid fuel cells.

UNIT - IV: NANOTECHNOLOGY

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

UNIT - V: LASER AND FIBER OPTICS

Lasers: Laser beam characteristics-three quantum processes-Einstein coefficients and their relations- lasing action - pumping methods- ruby laser, He-Ne laser , CO₂ laser, Argon ion Laser, Nd:YAG laser- semiconductor laser-applications of laser.

Fiber Optics: Introduction to optical fiber- advantages of optical Fibers - total internal reflection- construction of optical fiber - acceptance angle - numerical aperture- classification of optical fibers- losses in optical fiber - optical fiber for communication system - applications.

TEXT BOOKS:

1. M. N. Avadhanulu, P.G. Kshirsagar & TVS Arun Murthy” A Text book of Engineering Physics”- S. Chand Publications, 11th Edition 2019.
2. Engineering Physics by Shatendra Sharma and Jyotsna Sharma, Pearson Publication, 2019
3. Semiconductor Physics and Devices- Basic Principle – Donald A, Neamen, Mc Graw Hill, 4th Edition, 2021.
4. B.K. Pandey and S. Chaturvedi, Engineering Physics, Cengage Learning, 2nd Edition, 2022.
5. Essentials of Nanoscience & Nanotechnology by Narasimha Reddy Katta, Typical Creatives NANO DIGEST, 1st Edition, 2021.

REFERENCE BOOKS:

1. Quantum Physics, H.C. Verma, TBS Publication, 2nd Edition 2012.
2. Fundamentals of Physics – Halliday, Resnick and Walker, John Wiley & Sons, 11th Edition, 2018.
3. Introduction to Solid State Physics, Charles Kittel, Wiley Eastern, 2019.
4. Elementary Solid State Physics, S.L. Gupta and V. Kumar, Pragathi Prakashan, 2019.
5. A.K. Bhandhopadhyay - Nano Materials, New Age International, 1st Edition, 2007.
6. Energy Materials a Short Introduction to Functional Materials for Energy Conversion and Storage Aliaksandr S. Bandarenka, CRC Press Taylor & Francis Group
7. Energy Materials, Taylor & Francis Group, 1st Edition, 2022.

ME 203 ES: ENGINEERING WORKSHOP**B.Tech. I Year II Sem.****L T P C****0 13 2.5****Pre-requisites:** Practical skill**Course Objectives:**

- Familiarize students with basic workshop tools, their functions, and operational procedures, and provide practical experience in trades such as fitting, carpentry, foundry, and welding.
- Develop skills to fabricate components using sheet metal operations applicable in real-world engineering scenarios.
- Introduce basic electrical engineering principles through hands-on house wiring practice to ensure safe and functional electrical installations.
- Enable students to select and utilize appropriate tools and techniques for shaping, forming, and assembling metal components.
- Provide exposure to practical applications in plumbing, machine shop operations, advanced metal cutting methods (such as water plasma cutting), power tools, and woodworking used in construction and manufacturing industries.

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

- Identify Workshop tools and their operational capabilities & Practice on manufacturing of components using workshop trades including fitting, carpentry, and foundry and welding.
- Build different parts with metal sheets in real work applications.
- Apply basic electrical engineering knowledge for house wiring practice.
- Select appropriate tools and techniques for shaping and forming metal components.
- Demonstrate the Plumbing, Machine Shop, Metal Cutting (Water Plasma), Power tools in construction and Wood Working.

1. TRADES FOR EXERCISES:**At least two exercises from each trade:**

- I. Carpentry – (T-Lap Joint, Dovetail Joint, Mortise & Tenon Joint)
- II. Fitting – (V-Fit, Dovetail Fit & Semi-circular fit)
- III. Tin-Smithy – (Square Tin, Rectangular Tray & Conical Funnel)
- IV. Foundry – (Preparation of Green Sand Mould using Single Piece and Split Pattern)
- V. Welding Practice – (Arc Welding & Gas Welding)
- VI. House-wiring – (Parallel & Series, Two-way Switch and Tube Light)
- VII. Black Smithy – (Round to Square, Fan Hook and S-Hook)

2. TRADES FOR DEMONSTRATION & EXPOSURE:

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

EN 204 HS: ENGLISH FOR SKILL ENHANCEMENT**B.Tech. I Year II Sem.**

L	T	P	C
2	0	0	2

Course Objectives: This course will enable the students to:

- Build a strong foundation in word formation, sentence structure, and develop precision in both reading and writing practices.
- Enable students to recognize and apply grammatical structures effectively in both spoken and written communication, while enhancing critical reading sub-skills such as skimming, scanning, and inference.
- Train students to write clearly and efficiently in academic, business, and personal contexts using appropriate language, style, tone, and formatting conventions.
- Strengthen vocabulary for effective and engaging communication across diverse contexts, improving clarity and expression.
- Introduce students to technical vocabulary and develop their ability to synthesize ideas and information into structured, coherent, and professional written reports and documents.

Course Outcomes: Students will be able to:

- Demonstrate a clear understanding of word formation and develop the ability to read and write precisely
- Recognize and understand the meaning of targeted grammatical structures in written and spoken form and develop essential reading sub-skills
- Write efficiently for academic, business and personal situations using appropriate style and format
- Acquire strong vocabulary for clear and engaging expression of ideas
- Comprehend key technical terms and cultivate the ability to synthesize information into professional, well-organized reports

UNIT - I

Chapter entitled '*Toasted English*' by R.K.Narayan from "*English: Language, Context and Culture*" published by Orient BlackSwan, Hyderabad.

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

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

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

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

UNIT - II

Chapter entitled '*Appro JRD*' by Sudha Murthy from "*English: Language, Context and Culture*" published by Orient BlackSwan, Hyderabad.

Vocabulary: Words Often Misspelt - Homophones, Homonyms and Homographs

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

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

Writing: Nature and Style of Writing- Defining /Describing People, Objects, Places and Events – Classifying- Providing Examples or Evidence.

UNIT - III

Chapter entitled '*Lessons from Online Learning*' by F.Haider Alvi, Deborah Hurst et al from "*English: Language, Context and Culture*" published by Orient BlackSwan, Hyderabad.

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

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

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

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

UNIT - IV

Chapter entitled 'Art and Literature' by Abdul Kalam from "*English: Language, Context and Culture*" published by Orient BlackSwan, Hyderabad.

Vocabulary: Standard Abbreviations in English

Grammar: Redundancies and Clichés in Oral and Written Communication.

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

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

UNIT - V

Chapter entitled 'Go, Kiss the World' by Subroto Bagchi from "*English: Language, Context and Culture*" published by Orient BlackSwan, Hyderabad.

Vocabulary: Technical Vocabulary and their Usage

Grammar: Common Errors in English (*Covering all the other aspects of grammar which were not covered in the previous units*)

Reading: Reading Comprehension-Exercises for Practice

Writing: Technical Reports- Introduction – Characteristics of a Report – Categories of Reports
Formats- Structure of Reports (Manuscript Format) -Types of Reports - Writing a 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: 1.** 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.
- **Note: 2.** Based on the recommendations of NEP2020, teachers are requested to be flexible to adopt Blended Learning in dealing with the course contents. They are advised to teach 40 percent of each topic from the syllabus in blended mode.

TEXT BOOK:

1. "English: Language, Context and Culture" by Orient BlackSwan Pvt. Ltd, Hyderabad. 2022. Print.

REFERENCE BOOKS:

1. Effective Academic Writing by Liss and Davis (OUP)
2. Richards, Jack C. (2022) Interchange Series. Introduction, 1,2,3. Cambridge University Press
3. Wood, F.T. (2007). Remedial English Grammar. Macmillan.
4. Chaudhuri, Santanu Sinha. (2018). Learn English: A Fun Book of Functional Language, Grammar and Vocabulary. (2nd ed.,). Sage Publications India Pvt. Ltd.
5. (2019). Technical Communication. Wiley India Pvt. Ltd.
6. Vishwamohan, Aysha. (2013). English for Technical Communication for Engineering Students. Mc Graw-Hill Education India Pvt. Ltd.
7. Swan, Michael. (2016). Practical English Usage. Oxford University Press. Fourth Edition.

EC 205 ES: ELECTRONIC DEVICES AND CIRCUITS**B.Tech. I Year II Sem.****L T P C**
2 0 0 2**Course Objectives:**

- Develop a foundational understanding of word formation, enabling students to read critically and write with clarity and precision.
- Enhance grammatical accuracy and foster essential reading sub-skills such as prediction, inference, summarizing, and comprehension in both written and spoken contexts.
- Equip students to write effectively for academic, personal, and workplace-related scenarios using appropriate formats, tones, and organizational structures.
- Enrich vocabulary to improve the fluency, coherence, and impact of written and spoken communication across diverse situations.
- Introduce students to technical vocabulary and cultivate their ability to synthesize, organize, and present information in professional and well-structured reports.

Course Outcomes: Upon completion of the Course, the students will be able to:

- Demonstrate a clear understanding of word formation and develop the ability to read and write precisely
- Recognize and understand the meaning of targeted grammatical structures in written and spoken form and develop essential reading sub-skills
- Write efficiently for academic, business and personal situations using appropriate style and format
- Acquire strong vocabulary for clear and engaging expression of ideas
- Comprehend key technical terms and cultivate the ability to synthesize information into professional, well-organized reports

UNIT - I

Diodes: Diode - Static and Dynamic resistances, Equivalent circuit, Diffusion and Transition Capacitances, V-I Characteristics, Diode as a switch- switching times.

UNIT - II

Diode Applications: Rectifier - Half Wave Rectifier, Full Wave Rectifier, Bridge Rectifier, Rectifiers with Capacitive and Inductive Filters, Clippers-Clipping at two independent levels, Clamper-Clamping Circuit Theorem, Clamping Operation, Types of Clampers.

UNIT - III

Bipolar Junction Transistor (BJT): Principle of Operation, Common Emitter, Common Base and Common Collector Configurations, Transistor as a switch, switching times,

UNIT - IV

Junction Field Effect Transistor (FET): Construction, Principle of Operation, Pinch-Off Voltage, Volt- Ampere Characteristic, Comparison of BJT and FET, FET as Voltage Variable Resistor, MOSFET, MOSTET as a capacitor.

UNIT – V

Special Purpose Devices: Zener Diode - Characteristics, Zener diode as Voltage Regulator, Principle of Operation - SCR, Tunnel diode, UJT, Varactor Diode, Photo diode, Solar cell, LED, Schottky diode.

TEXT BOOKS:

1. Jacob Millman - Electronic Devices and Circuits, McGraw Hill Education
2. Robert L. Boylestead, Louis Nashelsky- Electronic Devices and Circuits theory, 11th Edition, 2009, Pearson.

REFERENCE BOOKS:

1. Horowitz -Electronic Devices and Circuits, David A. Bell – 5th Edition, Oxford.
2. Chinmoy Saha, Arindam Halder, Debaati Ganguly - Basic Electronics-Principles and Applications, Cambridge, 2018.

CS 206 ES: PYTHON PROGRAMMING LABORATORY**B.Tech. I Year II Sem.**

L	T	P	C
0	1	2	2

Course Objectives:

- Enable students to develop application-specific programs using Python with a focus on real-world problem-solving.
- Provide comprehensive understanding of core Python data structures such as strings, lists, tuples, and dictionaries for effective data manipulation.
- Promote good programming practices through modular programming techniques, file handling, and usage of the Python standard library for efficient and readable code.
- Introduce the integration of Python in implementing digital systems and simulations for basic engineering applications.
- Encourage the use of methods (functions) to modularize code, enhance reusability, and improve maintainability in program design.

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

- Develop the application specific codes using python.
- Understand Strings, Lists, Tuples and Dictionaries in Python
- Verify programs using modular approach, file I/O, Python standard library
- Implement Digital Systems using Python
- Able to modularize the code with Methods so that they can be reused

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

Week -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.
- Start a Python interpreter and use it as a Calculator.
- 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
- Read name, address, email and phone number of a person through keyboard and print the details.

Week - 2:

- Print the below triangle using for loop. 5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
- 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)
- Python Program to Print the Fibonacci sequence using while loop
- Python program to print all prime numbers in a given interval (use break)

Week - 3:

- i) Write a program to convert a list and tuple into arrays.
ii) Write a program to find common values between two arrays.
- Write a function called gcd that takes parameters a and b and returns their greatest common divisor.
- Write a function called palindrome that takes a string argument and returns True 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:

- 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.
- 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.

- i). 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.
- ii). The wordlist I provided, `words.txt`, doesn't contain single letter words. So you might want to add "I", "a", and the empty string.
- iii). 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. i) Add a comma between the characters. If the given word is 'Apple', it should become 'A,p,p,l,e'
- ii) Remove the given word in all the places in a string?
- iii) Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding upper case letter and the rest of the letters in the word by corresponding letters in lower case without using a built-in function?
4. Writes a recursive function that generates all binary strings of n-bit length

Week - 5:

1. i) Write a python program that defines a matrix and prints
- ii) Write a python program to perform addition of two square matrices
- iii) Write a python program to perform multiplication of two square matrices
2. How do you make a module? Give an example of construction of a module using different geometrical shapes and operations on them as its functions.
3. Use the structure of exception handling all general purpose exceptions.

Week-6:

1. a. Write a function called `draw_rectangle` that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas.
- b. Add an attribute named `color` to your Rectangle objects and modify `draw_rectangle` so that it uses the `color` attribute as the fill color.
- c. Write a function called `draw_point` that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas.
- d. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called `draw_circle` that draws circles on the canvas.
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- 7

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.
3. Write a Python code to Read text from a text file, find the word with most number of occurrences
4. 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 - 8:

1. Import numpy, Plotpy and Scipy and explore their functionalities.
2. a) Install NumPy package with pip and explore it.
3. Write a program to implement Digital Logic Gates – AND, OR, NOT, EX-OR
4. Write a program to implement Half Adder, Full Adder, and Parallel Adder
5. Write a GUI program to create a window wizard having two text labels, two text fields and two buttons as Submit and Reset.

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Python for Data Science, Dr. Mohd. Abdul Hameed, Wiley Publications - 1st Ed. 2021.
2. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
3. Python Programming A Modular Approach with Graphics, Database, Mobile, and Web Applications, Sheetal Taneja, Naveen Kumar, Pearson
4. Programming with Python, A User's Book, Michael Dawson, Cengage Learning, India Edition
5. Think Python, Allen Downey, GreenTea Press
6. Core Python Programming, W. Chun, Pearson
7. Introduction to Python, Kenneth A. Lambert, Cengage

PH 207 BS: APPLIED PHYSICS LABORATORY**B.Tech. I Year II Sem.**

L	T	P	C
0	0	3	1.5

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

1. Capable of handling instruments related to the Hall effect and photoelectric effect experiments and their measurements.
2. Understand the characteristics of various devices such as PN junction diode, Zener diode, BJT, LED, solar cell, lasers and optical fiber and measurement of energy gap and resistivity of semiconductor materials.
3. Able to measure the characteristics of dielectric constant of a given material.
4. Study the behavior of B-H curve of ferromagnetic materials.
5. Understanding the method of least squares fitting.

Course Outcomes: The students will be able to:

1. Know the determination of the Planck's constant using Photo electric effect and identify the material whether it is n-type or p-type by Hall experiment.
2. Appreciate quantum physics in semiconductor devices and optoelectronics.
3. Gain the knowledge of applications of dielectric constant.
4. Understand the variation of magnetic field and behavior of hysteresis curve.
5. Carried out data analysis.

LIST OF EXPERIMENTS:

1. Determination of work function and Planck's constant using photoelectric effect.
2. Determination of Hall co-efficient and carrier concentration of a given semiconductor.
3. Characteristics of series and parallel LCR circuits.
4. V-I characteristics of a p-n junction diode and Zener diode
5. Input and output characteristics of BJT (CE, CB & CC configurations)
6. a) V-I and L-I characteristics of light emitting diode (LED)
b) V-I Characteristics of solar cell
7. Determination of Energy gap of a semiconductor.
8. Determination of the resistivity of semiconductor by two probe method.
9. Study B-H curve of a magnetic material.
10. Determination of dielectric constant of a given material
11. a) Determination of the beam divergence of the given LASER beam
b) Determination of Acceptance Angle and Numerical Aperture of an optical fiber.
12. Understanding the method of least squares – torsional pendulum as an example.

*Note: Any 8 experiments are to be performed.***REFERENCE BOOK:**

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

EN 208 HS: ENGLISH LANGUAGE AND COMMUNICATION SKILLS LABORATORY**B.Tech. I Year II Sem.****L T P C**
0 0 2 1

The **English Language and Communication Skills (ELCS) Lab** focuses 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:

- Develop a clear understanding of word formation, grammar, and sentence construction to promote accuracy in reading and writing.
- Strengthen comprehension and application of targeted grammatical structures in both spoken and written forms, along with essential reading strategies such as skimming, scanning, and critical analysis.
- Train students to write effectively across academic, business, and personal contexts using appropriate styles, tones, and formats.
- Build a strong vocabulary base for clear, coherent, and engaging expression of thoughts and ideas in varied communication scenarios.
- Familiarize students with technical vocabulary and develop their ability to extract, synthesize, and present information in the form of professionally structured and well-organized written reports.

Course Outcomes: Students will be able to:

- Demonstrate a clear understanding of word formation and develop the ability to read and write precisely
- Recognize and understand the meaning of targeted grammatical structures in written and spoken form and develop essential reading sub-skills
- Write efficiently for academic, business and personal situations using appropriate style and format
- Acquire strong vocabulary for clear and engaging expression of ideas
- Comprehend key technical terms and cultivate the ability to synthesize information into professional, well-organized reports

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

- Computer Assisted Language Learning (CALL) Lab**
- Interactive Communication Skills (ICS) Lab**

Listening Skills:**Objectives**

1. To enable students develop their listening skills so that they may appreciate the role in the LSRW skills approach to language and improve their pronunciation
2. To equip students with necessary training in listening, so that they can comprehend the speech of people of different backgrounds and regions

Students should be given practice in listening to the sounds of the language, to be able to recognize them and find the distinction between different sounds, to be able to mark stress and recognize and use the right intonation in sentences.

- Listening for general content
- Listening to fill up information
- Intensive listening
- Listening for specific information

Speaking Skills:**Objectives**

1. To involve students in speaking activities in various contexts
2. To enable students express themselves fluently and appropriately in social and professional contexts
 - Oral practice
 - Describing objects/situations/people
 - Role play – Individual/Group activities
 - Just A Minute (JAM) Sessions

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

Exercise – I CALL**Lab:**

Understand: Listening Skill- Its importance – Purpose- Process- Types- Barriers- Effective Listening. *Practice:* Introduction to Phonetics – Speech Sounds – Vowels and Consonants – Minimal Pairs- Consonant Clusters- Past Tense Marker and Plural Marker- *Testing Exercises*

ICS Lab:

Understand: Spoken vs. Written language- Formal and Informal English.

Practice: Ice-Breaking Activity and JAM Session- Situational Dialogues – Greetings – Taking Leave – Introducing Oneself and Others.

Exercise – II**CALL Lab:**

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

Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms- Stress pattern in sentences – Intonation - *Testing Exercises*

ICS Lab:

Understand: Features of Good Conversation – Strategies for Effective Communication.

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

Exercise - III**CALL Lab:**

Understand: Errors in Pronunciation-Neutralising Mother Tongue Interference (MTI).

Practice: Common Indian Variants in Pronunciation – Differences between British and American Pronunciation -*Testing Exercises*

ICS Lab:

Understand: Descriptions- Narrations- Giving Directions and Guidelines – Blog Writing

Practice: Giving Instructions – Seeking Clarifications – Asking for and Giving Directions – Thanking and Responding – Agreeing and Disagreeing – Seeking and Giving Advice – Making Suggestions.

Exercise – IV**CALL Lab:**

Understand: Listening for General Details.

Practice: Listening Comprehension Tests - *Testing Exercises*

ICS Lab:

Understand: Public Speaking – Exposure to Structured Talks - Non-verbal Communication- Presentation Skills.

Practice: Making a Short Speech – Extempore- Making a Presentation.

Exercise – V**CALL Lab:**

Understand: Listening for Specific Details.

Practice: Listening Comprehension Tests -*Testing Exercises*

ICS Lab:

Understand: Group Discussion

Practice: Group Discussion

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 T. V. or LCD, a digital stereo –audio & video system and camcorder etc.

Source of Material (Master Copy):

- *Exercises in Spoken English. Part 1,2,3.* CIEFL and Oxford University Press

Note: Teachers are requested to make use of the master copy and get it tailor-made to suit the contents of the syllabus.

Suggested Software:

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

REFERENCE BOOKS:

1. (2022). *English Language Communication Skills – Lab Manual cum Workbook*. Cengage Learning India Pvt. Ltd.
2. Shobha, KN & Rayen, J. Lourdes. (2019). *Communicative English – A workbook*. Cambridge University Press
3. Kumar, Sanjay & Lata, Pushp. (2019). *Communication Skills: A Workbook*. Oxford University Press
4. Board of Editors. (2016). *ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities*. Orient Black Swan Pvt. Ltd.
5. Mishra, Veerendra et al. (2020). *English Language Skills: A Practical Approach*. Cambridge University Press.

CS 209 ES: IT WORKSHOP**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.

- Develop the ability to perform basic hardware troubleshooting and resolve common computer system issues.
- Provide an understanding of internal and external hardware components and their interdependencies for effective system management.
- Create awareness about computer security threats such as viruses and worms, and introduce methods for safeguarding systems.
- Enable students to create well-structured documents and effective presentations using standard office software.
- Equip students with skills to perform data analysis, create charts, and carry out complex calculations using spreadsheets for academic and professional applications.

Course Outcomes:

- Perform Hardware troubleshooting
- Understand Hardware components and inter dependencies
- Safeguard computer systems from viruses/worms
- Document/ Presentation preparation
- Perform calculations using spreadsheets

PC Hardware

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

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

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

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

Internet & World Wide Web

Task1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

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

Task 3: Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

Task 4: Cyber Hygiene: Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

LaTeX and WORD

Task 1 – Word Orientation: The mentor needs to give an overview of LaTeX and Microsoft (MS) office or equivalent (FOSS) tool word: Importance of LaTeX and MS office or equivalent (FOSS) tool 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

Powerpoint

Task 1: Students will be working on basic power point utilities and tools which help them create basic powerpoint 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.

REFERENCE BOOKS:

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*.

MC210: ENVIRONMENTAL SCIENCE**B.Tech. I Year II Sem.**

L	T	P	C
3	0	0	0

Course Objectives:

- Develop a comprehensive understanding of environmental concepts, ecosystems, and biodiversity and their interrelationship with human activities.
- Enable students to evaluate the impacts of both renewable and non-renewable natural resource utilization, highlighting the need for sustainable resource management.
- Provide knowledge about the importance of biodiversity, its classification, threats, and conservation strategies to maintain ecological balance.
- Familiarize students with various types of pollution, their effects, and the design and application of pollution control equipment and techniques.
- Promote awareness and responsibility towards environmental protection, climate change, and sustainable practices for a better future.

Course Outcomes:

- To develop knowledge of the concept of environment, ecosystem and biodiversity
- Understand positive and negative effects of different natural resources
- Describe biodiversity and conservation of biodiversity.
- Design control equipment for pollution control.
- To develop awareness about environmental protection and sustainability.

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, Field visits.

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 resources:** Forest resources, **Energy resources:** growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. 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. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **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:** Wastewater Treatment methods: Primary, secondary and Tertiary.

Overview of air pollution control technologies, Concepts of bioremediation. **Global Environmental Issues and Global Efforts:** Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

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 Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
- 2 Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

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

EC 301 PC: DIGITAL ELECTRONICS**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Course Objectives: This course aims

- To impart knowledge about the characteristics and functionalities of electronic components
- To develop the ability to design and analyze analog electronic circuits.
- To introduce the fundamentals of Boolean algebra and combinational logic design
- To provide knowledge on digital logic families and gate-level realization techniques
- To introduce memory elements and asynchronous sequential circuit design

Course Outcomes:

- Know the characteristics of various components and understand the utilization of components.
- Design and analyze small signal amplifier circuits.
- Learn Postulates of Boolean algebra to minimize combinational functions and, design and analyze combinational circuits
- Know about the logic families and realization of logic gates
- Understand the operation, characteristics of various memory types, design and analyze asynchronous sequential logic circuits

UNIT - I:

BOOLEAN ALGEBRA AND LOGIC GATES: Digital Systems, Binary Numbers, Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes, Binary Storage and Registers, Binary logic.

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

UNIT - II:

GATE – LEVEL MINIMIZATION: The map method, Four-variable map, Five-Variable map, product of sums simplification Don't-care conditions, NAND and NOR implementation other Two-level implementations, Exclusive – Or function.

UNIT - III:

COMBINATIONAL LOGIC: Combinational Circuits, Analysis procedure Design procedure, Binary Adder-Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

UNIT - IV:

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

UNIT - V

MEMORIES AND ASYNCHRONOUS SEQUENTIAL LOGIC: Introduction, Random-Access Memory, Memory Decoding, Error Detection and correction Read-only memory, Programmable logic Array programmable Array logic, Sequential Programmable Devices.

Introduction, Analysis Procedure, Circuits with Latches, Design Procedure, Reduction of state and Flow Tables, Race-Free state Assignment Hazards, Design Example.

TEXT BOOKS:

1. Digital Design – Third Edition, M. Morris Mano, Pearson Education/PHI.
2. Digital Principles and Applications Albert Paul Malvino Donald P. Leach TATA McGraw Hill Edition.
3. Fundamentals of Logic Design, Roth, 5th Edition, Thomson.

REFERENCE BOOKS:

1. Switching and Finite Automata Theory by Zvi. Kohavi, Tata McGraw Hill.
2. Switching and Logic Design, C.V.S. Rao, Pearson Education
3. Digital Principles and Design – Donald D.Givone, Tata McGraw Hill, Edition.
4. Fundamentals of Digital Logic and Microcomputer Design, 5TH Edition, M. Rafiquzzaman John Wiley.

CS 302 PC: DATA STRUCTURES**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Prerequisites: Programming for Problem Solving**Course Objectives**

- To introduce fundamental linear data structures and their applications
- To develop understanding of hashing techniques and dictionary implementations
- To explore advanced tree-based data structures for dynamic data operations
- To analyze and implement efficient algorithms for graphs and sorting
- To apply tree structures and algorithms for pattern matching and information retrieval

Course Outcomes

- Ability to select and implement basic linear data structures like linked lists, stacks, and queues for efficient data storage and retrieval.
- Ability to assess the efficiency of different dictionary implementations and understand various collision resolution techniques in hash tables.
- Design and implement different types of search trees, such as AVL trees, Red-Black trees, and Splay trees, and evaluate their performance for search, insertion, and deletion operations.
- Ability to implement and analyze graph algorithms and various sorting techniques, including heap sort and external merge sort, for different data scenarios.
- Apply pattern matching algorithms and tree structures to solve problems related to string processing and information retrieval.

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching. Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing- linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion , B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Definition, Insertion Operation Splay Trees, Definition, Insertion Operation.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.
Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

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

REFERENCE BOOK:

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

CS 303 PC: COMPUTER ORIENTED STATISTICAL METHODS**B.Tech. II Year I Sem.**

L	T	P	C
3	1	0	4

Pre-requisites: Mathematics courses of first year of study.**Course Objectives:** To learn

- To introduce foundational concepts of probability and its application to real-world case studies
- To develop the ability to compute mathematical expectations and apply discrete probability distributions
- To explore continuous probability distributions and sampling techniques
- To equip students with tools for statistical inference
- To introduce the theory and application of stochastic processes and Markov chains

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

- Apply the concepts of probability and distributions to some case studies
- Ability to compute and interpret the mathematical expectation, and apply discrete probability distributions like Binomial, Geometric, and Poisson to real-world scenarios.
- Ability to apply continuous probability distributions, and understand the importance of sampling distributions.
- Ability to perform statistical inference through estimation and hypothesis testing.
- Ability to model and analyze stochastic processes, Markov chains

UNIT - I: Probability**10 L**

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

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

UNIT - II: Expectation and discrete distributions**10 L**

Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem.

Discrete Probability Distributions: Binomial Distribution, Poisson distribution.

UNIT - III: Continuous and Sampling Distributions**10 L**

Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial Distributions.

Fundamental Sampling Distributions: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, t - Distribution, F- Distribution.

UNIT - IV: Sample Estimation & Tests of Hypotheses**10 L**

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

Statistical Hypotheses: General Concepts, Testing a Statistical Hypothesis, Single sample: Tests concerning a single mean, Two samples: tests on two means, One sample: test on a single proportion. Two samples: tests on two proportions, Two- sample tests concerning variances.

UNIT-V: Stochastic Processes and Markov Chains**8L**

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.

TEXT BOOKS:

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, Probability & Statistics for Engineers & Scientists, 9th Ed. Pearson Publishers.

2. S C Gupta and V K Kapoor, Fundamentals of Mathematical statistics, Khanna publications.
3. S.D.Sharma, Operations Research, Kedarnath and Ramnath Publishers, Meerut, Delhi.

REFERENCE BOOKS:

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

CS 304 PC: COMPUTER ORGANIZATION AND ARCHITECTURE**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Co-requisite: A Course on “Digital Electronics”.**Course Objectives**

- To introduce the fundamentals of instruction set architecture and its influence on processor design
- To develop knowledge of the design and functioning of core components in a computer system
- To equip students with the ability to interpret and operate on data representations used in computers
- To assess and analyze design trade-offs in processor and memory system architecture
- To introduce the concept of instruction pipelining and strategies to optimize execution

Course Outcomes

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

UNIT - I

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

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

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

UNIT - II

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

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

UNIT - III

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

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

UNIT - IV

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

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

UNIT - V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

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

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

TEXT BOOK:

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

REFERENCE BOOKS:

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

CS 305 PC: OBJECT ORIENTED PROGRAMMING THROUGH JAVA**B.Tech. II Year I Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- To introduce the basic programming constructs such as control structures, constructors, string handling, and garbage collection in Java for building foundational programming skills.
- To impart knowledge on object-oriented programming concepts, focusing on inheritance types like multilevel, hierarchical, and multiple using extends and implements keywords.
- To provide understanding of multithreading mechanisms and their applications in enabling inter-process communication for efficient concurrent programming.
- To familiarize students with Graphical User Interface (GUI) design, using Abstract Window Toolkit (AWT) and Swing components for developing user-friendly applications.
- To enable students to develop and deploy Java applets, allowing interaction with client environments and integration with server-side systems.

Course Outcomes

- Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
- Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords
- Use multithreading concepts to develop inter process communication.
- Understand the process of graphical user interface design and implementation using AWT orSwings
- 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. A way of viewing world – Agents, responsibility, messages, methods, 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, method binding, inheritance, overriding and exceptions, 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 uses, using final 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. Exploring java.io.

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. String handling, Exploring java.util. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads. Enumerations, autoboxing, annotations, generics.

UNIT - IV

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes. The AWT class hierarchy, user interface components- labels, button, canvas, scrollbars, text components, check box, checkbox groups, choices, lists panels – scrollpane, dialogs, menubar, graphics, layout manager – layout manager types – border, grid, flow, card and grid bag.

UNIT - V

Applets – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets. Swing – Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

TEXT BOOKS:

1. Java the complete reference, 7th edition, Herbert schildt, TMH.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

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

CS 306 PC: DATA STRUCTURES LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	3	1.5

Prerequisites: A Course on “Programming for problem solving”.**Course Objectives:**

- To introduce the concepts of dynamic memory management and linked data structures
- To develop skill in implementing stacks and queues using arrays and pointers
- To impart knowledge of sorting algorithms and their implementation in C
- To introduce fundamental searching techniques for data retrieval
- To introduce basic graph algorithms and string pattern matching techniques

Course Outcomes:

- Ability to develop C programs for implementing and manipulating various types of linked lists.
- Ability to implement stack and queue data structures using arrays and pointers
- Ability to implement sorting algorithms
- Ability to implement searching algorithms
- Ability to implement tgraph traversal algorithms and pattern matching techniques

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) Pointers
5. Write a program that implement Queue (its operations) using
i) Arrays ii) Pointers
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order
i) Quick sort ii) Heap sort iii) Merge 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. Implement a Pattern matching algorithms using Boyer- Moore, Knuth-Morris-Pratt

TEXT BOOKS:

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

REFERENCE BOOK:

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

CS 307 PC: OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	3	1.5

Course Objectives:

- To enable students to solve real-world problems using the Java Collection Framework
- To impart knowledge on designing reusable components using abstract classes and interfaces
- To introduce the concepts and implementation of multithreading in Java
- To train students in designing interactive user interfaces using Java Swing
- To strengthen problem-solving skills through implementation of data structures and algorithms in Java

Course Outcomes:

- Able to write programs for solving real world problems using the java collection framework
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.
- Able to implement data structures and algorithms in Java for efficient problem-solving.

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 Value and returns it in another text field, when the button named "Compute" is clicked.
4. 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.
5. 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.
6. 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.
7. 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.

8. 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.
9. 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.
Write a java program to display the table using Labels in Grid Layout.
10. 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).
11. 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).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of inter thread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

REFERENCE BOOKS:

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

CS 308 PC: DATA VISUALIZATION - R PROGRAMMING/ POWER BI**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	1

Course Objectives:

- To introduce students to the process of importing and preparing data in Tableau
- To develop a foundational understanding of Tableau's core concepts: Dimensions and Measures
- To teach visualization design through programmatic thinking and visual mapping
- To guide students in building interactive dashboards
- To apply graphical user interfaces and dashboards to solve real-world problems

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

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

Lab Problems:

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

REFERENCE BOOKS:

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

MC309: GENDER SENSITIZATION LAB**B.Tech. II Year I Sem.**

L	T	P	C
0	0	2	0

COURSE DESCRIPTION

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

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

Course Objectives

- To introduce students to foundational gender concepts and terminology
- To raise awareness of gender discrimination and promote gender equity
- To explore the intersection of gender with labor, politics, and economics
- To educate students on gender-based violence and human rights-based interventions
- To cultivate critical thinking on gender representation in media and literature

Course Outcomes

- Students will be able to define and articulate basic gender concepts and terminology and the processes of socialization that shape gender roles.
- Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
- Students will acquire insight into the gendered division of labour and its relation to politics and economics.
- Students will develop an understanding of the various types of gender-based violence, and will learn strategies for addressing and preventing such violence from a human rights perspective.
- Students will critically assess gender representation in media and literature, and advocate for gender-sensitive language and equal relationships.

Unit-I: UNDERSTANDING GENDER

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

Unit – II: GENDER ROLES AND RELATIONS

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

Unit – III: GENDER AND LABOUR

Division and Valuation of Labour-Housework: The Invisible Labor- “My Mother doesn’t Work.” “Share the Load.”-Work: Its Politics and Economics -Fact and Fiction. Unrecognized and Unaccounted work. -Gender Development Issues-Gender, Governance and Sustainable Development-Gender and Human Rights-Gender and Mainstreaming

Unit – IV: GENDER - BASED VIOLENCE

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

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

Unit – V: GENDER AND CULTURE

Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular

Literature- Gender Development Issues-Gender Issues-Gender Sensitive Language-Gender and Popular

Literature - Just Relationships: Being Together as Equals

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

Note: Since it is Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field from engineering departments.

- **Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work and in-class assignments. Apart from the above prescribed book, Teachers can make use of any authentic materials related to the topics given in the syllabus on “Gender”.**
- **ESSENTIAL READING:** The Textbook, “*Towards a World of Equals: A Bilingual Textbook on Gender*” written by A.Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu **published by Telugu Akademi, Telangana Government in 2015.**

ASSESSMENT AND GRADING:

- Discussion & Classroom Participation: 20%
- Project/Assignment: 30%
- End Term Exam: 50%

CS 401 ES: DISCRETE MATHEMATICS**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives:

- To develop the ability to understand and construct rigorous mathematical proofs
- To introduce logic and set theory as tools for precise problem formulation
- To provide techniques for solving combinatorial and counting problems
- To teach the concepts of sequences and recurrence relations
- To apply graph theory concepts in computing contexts

Course Outcomes:

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

UNIT - I

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

UNIT - II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT - III

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

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 Subgraphs, 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 I. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCE BOOKS:

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.

MB 402 HS: BUSINESS ECONOMICS AND FINANCIAL ANALYSIS**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Course Objective: To learn the basic business types, impact of the economy on Business and Firms specifically.

- To introduce the fundamentals of business organizations and economic environment
- To develop a strong foundation in microeconomics and macroeconomics
- To analyze consumer behavior, market demand, and forecasting techniques
- To evaluate production and cost concepts in relation to different market structures
- To impart financial accounting knowledge and tools for financial decision-making

Course Outcome:

- Students will understand the various forms of business entities, the sources of capital, and the significance of key economic variables, including national income, inflation, and the business cycle.
- Students will grasp micro and macroeconomic concepts, national income, inflation, money supply, and business cycles.
- Analyze demand and supply elasticity, factors influencing them, and methods of demand forecasting.
- Evaluate production functions, cost types, market structures, and pricing strategies.
- Students will master financial accounting basics, prepare financial statements, and perform financial ratio analysis.

Unit – I: Introduction to Business and Economics

Business: Structure of Business Firm, Theory of Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for a Company, Non-Conventional Sources of Finance.

Economics: Significance of Economics, Micro and Macro Economic Concepts, Concepts and Importance of National Income, Inflation, Money Supply and Inflation, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Role of Business Economist, Multidisciplinary nature of Business Economics.

UNIT - II: Demand and Supply Analysis

Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

Supply Analysis: Determinants of Supply, Supply Function and Law of Supply.

UNIT - III: Production, Cost, Market Structures & Pricing

Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale, Different Types of Production Functions.

Cost analysis: Types of Costs, Short run and Long run Cost Functions.

Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Monopolistic Competition. **Pricing:** Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis, Cost Volume Profit Analysis.

UNIT - IV: Financial Accounting: Accounting concepts and Conventions, Accounting Equation, Double-Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts (Simple Problems).

UNIT - V: Financial Ratios Analysis: Concept of Ratio Analysis, Importance and Types of Ratios, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios – Analysis and Interpretation (simple problems).

TEXT BOOKS:

1. D. D. Chaturvedi, S. L. Gupta, Business Economics - Theory and Applications, International Book

House Pvt. Ltd. 2013.

2. Dhanesh K Khatri, Financial Accounting, Tata Mc –Graw Hill, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury, Managerial Economics, 2e, Tata Mc Graw Hill Education Pvt. Ltd. 2012.

REFERENCE BOOKS:

1. Paresh Shah, Financial Accounting for Management 2e, Oxford Press, 2015.
2. S. N. Maheshwari, Sunil K Maheshwari, Sharad K Maheshwari, Financial Accounting, 5e, Vikas Publications, 2013.

CS 403 PC: OPERATING SYSTEMS**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:

- To introduce the fundamentals of operating system concepts and system components
- To provide knowledge of process scheduling and system-level programming
- To explain process synchronization and interprocess communication
- To educate memory management and page replacement techniques
- To familiarize students with file system architecture and operations

Course Outcomes:

- Able to control access to a computer and the files that may be shared and demonstrate the knowledge of the components of computer and their respective roles in computing.
- Understand CPU scheduling criteria and algorithms, and effectively utilize programming languages, operating systems, and architectures.
- Understand synchronization mechanisms and inter process communication mechanisms.
- Ability to understand and implement various memory management techniques and algorithms for page replacement
- Ability to understand file system operations and file system-related system calls
-

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.

REFERENCE BOOKS:

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

CS 404 PC: DATABASE MANAGEMENT SYSTEMS**B.Tech. II Year II Sem.****L T P C****3 0 0 3****Prerequisites:** A course on “Data Structures”.**Course Objectives:**

- To introduce the fundamental concepts of database systems and data modelling
- To develop an understanding of the relational data model and query formulation
- To teach the use of Structured Query Language (SQL) for data manipulation and retrieval
- To familiarize students with transaction management and concurrency control
- To provide knowledge of database storage, indexing, and access methods

Course Outcomes:

- Gain knowledge of fundamentals of DBMS, database design and ER diagrams.
- Understand the relational model, integrity constraints, relational algebra, and query languages.
- Master the basics of Structured Query Language (SQL) for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques

UNIT - I

Introduction to Databases and Database Management System: Database system Applications, Advantages of DBMS over File System, the Data Model, Levels of Abstraction in a DBMS, Database Languages –DDL, DML, DCL, Data Independence, Structure of a DBMS

Introduction to Database Design: Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Constraints, Keys, Entity-Relationship Diagram, Additional Features of the ER Model, Database Design with 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.

Introduction to SQL: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers, cursors

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, First, Second, Third normal forms, BCNF, lossless join decomposition, multivalued dependencies, Fourth normal form, Fifth normal form.

UNIT - IV

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

UNIT - V

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

B+ Trees: A Dynamic Index Structure.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 405 PC: SOFTWARE ENGINEERING**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- To provide knowledge of requirement elicitation and documentation techniques
- To introduce software architecture and design principles
- To educate architectural modeling and design methodologies
- To create awareness of software testing techniques and reporting
- To introduce software process and product measurement concepts

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.
- Ability to create an architectural design using UML and other design methodologies.
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report
- Ability to understand software measurement and metrics for both process and product

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI).

Process models: The waterfall model, Spiral model and Agile methodology

UNIT - II

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

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

UNIT - III

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

UNIT - IV

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

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

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM.

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 406 PC: OPERATING SYSTEMS LAB**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

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

Co-requisite: A course on “Operating Systems”.

Course Objectives:

- To develop practical skills in implementing CPU scheduling algorithms
- To introduce Unix system calls for basic file and I/O operations
- To implement deadlock avoidance and prevention techniques in practice
- To provide an understanding of synchronization mechanisms using semaphores
- To train students in Inter-Process Communication (IPC) and memory management techniques

Course Outcomes:

- Able to implement CPU Scheduling algorithms such as FCFS, SJF, Round Robin, and priority scheduling.
- Able to implement C programs using Unix system calls for I/O operations.
- Able to implement the Banker's Algorithm for Deadlock Avoidance and Prevention in C.
- Able to implement the Producer-Consumer problem using semaphores and Unix system calls.
- Able to implement Inter-Process Communication (IPC) mechanisms using C programs: Pipes, FIFOs, Message Queues, Shared Memory and memory management techniques in C.

List of Experiments:

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

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 407 PC: DATABASE MANAGEMENT SYSTEMS LAB**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

Co-requisites: “Database Management Systems”**Course Objectives:**

- To impart knowledge on database design principles and normalization techniques
- To develop proficiency in using SQL for data definition and manipulation
- To enhance problem-solving skills using advanced SQL queries
- To introduce procedural extensions of SQL using stored procedures and triggers
- To teach the use of cursors for row-by-row processing in database applications

Course Outcomes:

- Design database schema for a given application and apply normalization
- Acquire skills in using SQL commands for data definition and data manipulation
- Develop solutions for database applications using SQL queries
- Develop solutions for database applications using procedures and triggers
- Develop solutions for database applications using cursors

List of Experiments:

1. Draw an ER diagram for University Database, Company Database
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.

REFERENCE BOOKS:

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

CS 409 PC: NODE JS/ REACT JS/ DJANGO**B.Tech. II Year II Sem.**

L	T	P	C
0	0	2	1

Prerequisites: Object Oriented Programming through Java, HTML Basics**Course Objectives:**

- To implement the static web pages using HTML and do client side validation using JavaScript.
- To design and work with databases using Java
- To develop an end to end application using java full stack.
- To introduce Node JS implementation for server side programming.
- To experiment with single page application development using React.

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

- Build a custom website with HTML, CSS, and Bootstrap and little JavaScript.
- Demonstrate Advanced features of JavaScript and learn about JDBC
- Develop Server – side implementation using Java technologies
- Develop the server – side implementation using Node JS.
- Design a Single Page Application using React.

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 responsive web application using Bootstrap framework.
3. Use JavaScript for doing client – side validation of the pages implemented in experiment 1 and experiment 2.
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. Develop a java stand alone application that connects with the database (Oracle / mySql) and perform the CRUD operation on the database tables.
6. Create an xml for the bookstore. Validate the same using both DTD and XSD.
7. Design a controller with servlet that provides the interaction with application developed in experiment 1 and the database created in experiment 5.
8. Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)
9. Create a custom server using http module and explore the other modules of Node JS like OS, path, event.
10. Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)
11. For the above application create authorized end points using JWT (JSON Web Token).
12. Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages.
13. 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
14. Create a TODO application in react with necessary components and deploy it into github.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O'Reilly Media, 2nd Edition, 2008.
3. Vasanth Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

MC410: CONSTITUTION OF INDIA**B.Tech. II Year II Sem.**

L	T	P	C
3	0	0	0

Course Objectives: Students will be able:

- To examine the historical evolution of civil rights movements in pre-Gandhian India
- To familiarize students with the philosophical and intellectual foundations of social reform in India
- To provide knowledge of Fundamental Rights, Directive Principles, and Fundamental Duties
- To impart understanding of the structure and working of the Indian Parliament and Executive
- To introduce the role of local governance and electoral systems in Indian democracy

Course Outcomes: Students will be able to:

- Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.
- Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
- Understand the Fundamental Rights guaranteed by the Indian Constitution and the Directive Principles of State Policy and Fundamental Duties outlined in the Indian Constitution
- Understand the structure and functioning of the Parliament, roles, powers, functions of the Executive branch, and appointment and transfer of judges
- Understand the role and importance of Local Administration and the Election Commission and Welfare Institutions

Unit - 1 History of Making of the Indian Constitution- History of Drafting Committee.**Unit - 2** Philosophy of the Indian Constitution- Preamble Salient Features**Unit - 3** Contours of Constitutional Rights & Duties - Fundamental Rights

- Right to Equality
- Right to Freedom
- Right against Exploitation
- Right to Freedom of Religion
- Cultural and Educational Rights
- Right to Constitutional Remedies
- Directive Principles of State Policy
- Fundamental Duties.

Unit - 4 Organs of Governance: Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions**Unit - 5** Local Administration: District's Administration head: Role and Importance, Municipalities: Introduction, Mayor and role of Elected Representative, CEO of Municipal Corporation. Panchayat raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO ZilaPanchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy**Unit - 6** Election Commission: Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners. State Election Commission: Role and Functioning. Institute and Bodies for the welfare of SC/ST/OBC and women.**Suggested Reading:**

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

CS 501 PC: DESIGN AND ANALYSIS OF ALGORITHMS**B.Tech. III Year I Sem.****L T P C****3 1 0 4****Prerequisites:**

1. A course on “Computer Programming and Data Structures”.
2. A course on “Advanced Data Structures”.

Course Objectives:

- To introduce the foundations for analyzing algorithm efficiency
- To Enable students to choose appropriate data structures and algorithmic techniques
- To understand the impact of design choices on algorithm performance.
- To explain the principles and applications of the greedy method
- To explore computational complexity and intractable problems

Course Outcomes:

- Ability to analyze the performance of algorithms
- Ability to choose appropriate data structures and algorithm design methods for a specified application
- Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs
- Ability to understand the principles of the greedy method and its applications.
- Ability to comprehend the concepts of NP-Hard and NP-Complete problems and their implications on algorithm design and analysis.

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-queen’s 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 Graphs, Connected components, Biconnected components.

UNIT - V

Branch and Bound: General method, applications - Traveling 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 Rajasekharan, University press, 1998.

REFERENCE BOOKS:

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.

CS 502 PC: COMPUTER NETWORKS**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites**

1. A course on “Programming for problem solving”
2. A course on “Data Structures”

Course Objectives

- To provide foundational knowledge of computer networking concepts and technologies
- To develop a clear understanding of the layered architecture of OSI and TCP/IP models
- To impart practical knowledge of IP addressing, subnetting, and routing
- To familiarize students with core network protocols and their applications
- To introduce key application layer protocols and services

Course Outcomes

- Gain the knowledge of the basic computer network technology
- Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
- Obtain the skills of subnetting and routing mechanisms.
- Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.
- Understanding of key application layer protocols such as DNS, SNMP, SMTP, HTTP, and streaming audio and video.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission. Data link layer: Design issues, framing, Error detection and correction.

UNIT - II

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V

Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

REFERENCE BOOKS:

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

CS 503 PC: DEVOPS**B.Tech. III Year I Sem.****L T P C****3 0 0 3****Pre-Requisites:**

1. Software Engineering
2. Software Project Management

Course Objectives:

- To introduce the fundamental components of the DevOps environment
- To explore software development models and DevOps architectures
- To equip students in using version control and collaboration tools
- To guide students in the implementation of automated build systems and CI/CD pipeline.
- To introduce testing and deployment strategies in DevOps

Course Outcomes:

- Understand the various components of DevOps environment
- Identify Software development models and architectures of DevOps
- Utilize various source code management systems and project management tools to manage code repositories and facilitate collaboration.
- Implement and manage build systems and build pipelines to ensure efficient and reliable integration and delivery processes.
- Select an appropriate testing tool and deployment model for project

UNIT-I**Introduction to DevOps:**

Introduction, Agile development model, DevOps and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, identifying bottlenecks.

UNIT-II**Software development models and DevOps:**

DevOps Lifecycle for Business Agility, DevOps, and Continuous Testing. DevOps influence on Architecture: Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Micro services and the data tier, DevOps, architecture, and resilience.

UNIT-III**Introduction to project management:**

The need for source code control, the history of source code management, Roles and code, source code management system and migrations, shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

UNIT-IV**Integrating the system:**

Build systems, Jenkins build server, Managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, Building by dependency order, Build phases, Alternative build servers, Collating quality measures.

UNIT-V**Testing Tools and Deployment:**

Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, Testing backend integration points, Test-driven development, REPL-driven development. Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker.

TEXT BOOKS:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.

REFERENCE BOOKS:

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.

2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley

CS 509 PE: QUANTUM COMPUTING (Professional Elective – I)**B.Tech. III Year I Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- Introduce the fundamentals of quantum computing
- Develop the ability to apply core concepts of linear algebra, quantum mechanics, and molecular biology to understand and analyze quantum measurements and system correlations.
- Explore the physical realization of qubits through different quantum systems and technologies.
- Understand and implement basic quantum algorithms such as Deutsch-Jozsa, Grover's, and Shor's algorithms using appropriate frameworks and simulators.
- Examine the implications of quantum computing in cybersecurity, particularly its impact on classical cryptographic systems and the development of quantum-safe protocols.

Course Outcomes

- Understand basics of quantum computing
- Understand and apply fundamental concepts of linear algebra, quantum mechanics, and molecular biology to analyze measurements and correlations in quantum biological systems.
- Understand physical implementation of Qubit
- Understand Quantum algorithms and their implementation
- Understand The Impact of Quantum Computing on Cryptography

UNIT - I

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT - II

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements. **Background Physics:** Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. **Background Biology:** Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT - III

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT - IV

Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT - V

Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. **Quantum Information and Cryptography:** Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation

TEXT BOOKS:

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

REFERENCE BOOKS:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

CS 510 PE: ADVANCED COMPUTER ARCHITECTURE (Professional Elective – I)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites:** Computer Organization**Course Objectives:**

- Introduce students to various models of parallel computing to apply program partitioning, scheduling, and interconnection techniques for efficient parallel execution.
- Enable students to assess and optimize the performance of parallel systems using standard metrics, speedup laws, and scalability analysis, while understanding the significance of advanced hardware technologies and memory hierarchies.
- Develop a comprehensive understanding of shared-memory systems, consistency models, and advanced pipelining techniques, facilitating the design of high-performance instruction and arithmetic pipelines.
- Familiarize students with scalable architectures and multiprocessor interconnection mechanisms.
- Provide insights into vector and multivector processing techniques, allowing students to critically evaluate the architecture and efficiency of SIMD computers.

Course Outcomes:

- Understand and analyze various models of parallel computing to enable parallelism in programs.
- Evaluate parallel processing system performance using metrics and speedup laws, and examine the role of advanced hardware and memory hierarchies in achieving scalable parallel performance.
- Analyze shared-memory architectures, consistency models, and pipelining techniques, and design efficient instruction and arithmetic pipelines using linear and non-linear methods.
- Examine scalable architectures, multiprocessor interconnects, cache coherence protocols, and synchronization techniques in multicomputers and SIMD systems to support effective parallel computation.
- Explore vector processing principles and assess the organization and performance of multivector and SIMD systems, including case studies such as the Connection Machine CM-5.

UNIT - I

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

UNIT - II

Principles of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors

UNIT - III

Shared-Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

UNIT - IV

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivector and SIMD computers.

UNIT - V

Vector Processing Principles, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5.

TEXT BOOK

1. Advanced Computer Architecture, Kai Hwang, 2nd Edition, Tata McGraw Hill Publishers.

REFERENCE BOOKS:

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4th Edition, ELSEVIER.
2. Advanced Computer Architectures, S.G.Shiva, Special Indian edition, CRC, Taylor & Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.
5. Computer Architecture, B. Parhami, Oxford Univ. Press.

CS 511 PE: DATA ANALYTICS (Professional Elective – I)**B.Tech. III Year I Sem.**

L	T	P	C
3	0	0	3

Prerequisites

1. A course on “Database Management Systems”.
2. Knowledge of probability and statistics.

Course Objectives:

- To enable students to identify structured, semi-structured, and unstructured data sources.
- To provide students with insights into how data-driven analysis supports strategic planning, in business environments.
- Provide students with insights into how data-driven analysis supports operational efficiency, and competitive advantage in business environments.
- To equip students with knowledge of advanced techniques such as regression, classification, clustering, and time series analysis using modern tools and platforms.
- To enable effective data visualization and communication of analytical findings

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

- Understand various Data Sources and design Data Architecture
- Understand the impact of data analytics for business decisions and strategy
- Carry out data analysis/statistical analysis
- Carry out advanced data analysis techniques
- To carry out standard data visualization and formal inference procedures

UNIT - I

Data Management: Design Data Architecture and manage the data for analysis, understand various sources of Data like Sensors/Signals/GPS etc. Data Management, Data Quality(noise, outliers, missing values, duplicate data) and Data Processing & Processing.

UNIT - II

Data Analytics: Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and Variables, Data Modeling Techniques, Missing Imputations etc. Need for Business Modeling.

UNIT - III

Regression – Concepts, Blue property assumptions, Least Square Estimation, Variable Rationalization, and Model Building etc.

Logistic Regression: Model Theory, Model fit Statistics, Model Construction, Analytics applications to various Business Domains etc.

UNIT - IV

Object Segmentation: Regression Vs Segmentation – Supervised and Unsupervised Learning, Tree Building – Regression, Classification, Overfitting, Pruning and Complexity, Multiple Decision Trees etc. **Time Series Methods:** Arima, Measures of Forecast Accuracy, STL approach, Extract features from generated model as Height, Average Energy etc and Analyze for prediction

UNIT - V

Data Visualization: Pixel-Oriented Visualization Techniques, Geometric Projection Visualization Techniques, Icon-Based Visualization Techniques, Hierarchical Visualization Techniques, Visualizing Complex Data and Relations.

TEXT BOOKS:

1. Student’s Handbook for Associate Analytics – II, III.
2. Data Mining Concepts and Techniques, Han, Kamber, 3rd Edition, Morgan Kaufmann Publishers.

REFERENCE BOOKS:

1. Introduction to Data Mining, Tan, Steinbach and Kumar, Addison Wesley, 2006.
2. Data Mining Analysis and Concepts, M. Zaki and W. Meira
3. Mining of Massive Datasets, Jure Leskovec Stanford Univ. Anand Rajaraman Millway Labs Jeffrey D Ullman Stanford Univ.

CS 512 PE: IMAGE PROCESSING (Professional Elective – I)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites**

1. Students are expected to have knowledge in linear signals and systems, Fourier Transform, basic linear algebra, basic probability theory and basic programming techniques; knowledge of digital signal processing is desirable.
2. A course on “Computational Mathematics”
3. A course on “Computer Oriented Statistical Methods”

Course Objectives

- Introduce the fundamental concepts of two-dimensional signal acquisition, image sampling, and quantization processes.
- Familiarize students with spatial and frequency domain filtering techniques for image enhancement and analysis.
- Develop the ability to apply image restoration algorithms to recover degraded images using mathematical and interactive methods.
- Enable students to implement image segmentation techniques for identifying and extracting meaningful structures in digital images.
- Equip students with the knowledge to analyze and apply image compression methods by exploring redundancy reduction, fidelity measures, and designing both error-free and lossy compression systems.

Course Outcomes

- Demonstrate the knowledge of the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
- Demonstrate the knowledge of filtering techniques.
- Apply various image restoration to reconstruct degraded images using algebraic and interactive approaches.
- Implement image segmentation techniques to isolate meaningful structures in images.
- Analyze image compression methods by identifying types of redundancies, applying fidelity criteria, and differentiating between error-free and lossy compression models including the design of source encoders and decoders.

UNIT - I

Digital Image Fundamentals: Digital Image through Scanner, Digital Camera. Concept of Gray Levels. Gray Level to Binary Image Conversion. Sampling and Quantization. Relationship between Pixels. Imaging Geometry. 2D Transformations-DFT, DCT, KLT and SVD.

UNIT - II

Image Enhancement in Spatial Domain Point Processing, Histogram Processing, Spatial Filtering, Enhancement in Frequency Domain, Image Smoothing, Image Sharpening.

UNIT - III

Image Restoration Degradation Model, Algebraic Approach to Restoration, Inverse Filtering, Least Mean Square Filters, Constrained Least Squares Restoration, Interactive Restoration.

UNIT - IV

Image Segmentation Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region Oriented Segmentation.

UNIT - V

Image Compression Redundancies and their Removal Methods, Fidelity Criteria, Image Compression Models, Source Encoder and Decoder, Error Free Compression, Lossy Compression.

TEXT BOOK:

1. Digital Image Processing: R.C. Gonzalez & R. E. Woods, Addison Wesley/ Pearson Education, 2nd Ed, 2004.

REFERENCE BOOKS:

1. Fundamentals of Digital Image Processing: A. K. Jain, PHI.
2. Digital Image Processing using MATLAB: Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins: Pearson Education India, 2004.
3. Digital Image Processing: William K. Pratt, John Wiley, 3rd Edition, 2004.

CS 513 PE: PRINCIPLES OF PROGRAMMING LANGUAGES (Professional Elective – I)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites**

1. A course on “Mathematical Foundations of Computer Science”.
2. A course on “Computer Programming and Data Structures”.

Course Objectives

- Develop the ability to describe the syntax and semantics of programming languages using formal notation.
- Understand and analyze key programming language concepts such as names, bindings, scopes, data types, control structures, and type systems.
- Explore the design, implementation, and use of subprograms and abstract data types to support modular and maintainable code.
- Introduce and evaluate the concepts of concurrency, synchronization, and exception/event handling mechanisms used in contemporary programming languages.
- Compare different programming paradigms including functional, logic, and scripting languages, and understand their applications relative to imperative programming.

Course Outcomes

- Acquire the skills for expressing syntax and semantics in formal notation
- Analyze the concepts of names, bindings, scopes, and various data types, and apply control structures and type-checking mechanisms in different programming languages.
- Evaluate the design and implementation of subprograms and abstract data types for efficient modular programming.
- Demonstrate understanding of concurrency concepts, synchronization mechanisms, and exception/event handling techniques in modern programming languages.
- Compare and contrast paradigms of functional, logic, and scripting languages with imperative languages, and explore language-specific features

UNIT - I

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

UNIT - II

Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants

Data Types: Introduction, Primitive Data Types, Character String Types, User Defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence Expressions and Statements, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode

Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

UNIT - III

Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, **Coroutines Implementing Subprograms:** General Semantics of

Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping

Abstract Data Types: The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations

UNIT - IV

Concurrency: Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency.

Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

UNIT - V

Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages

Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.

Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2)

TEXT BOOKS:

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

REFERENCE BOOKS:

1. Programming Languages, 2nd Edition, A.B. Tucker, R. E. Noonan, TMH.
2. Programming Languages, K. C. Loudon, 2nd Edition, Thomson, 2003.

CS 514 PE: COMPUTER GRAPHICS (Professional Elective – II)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites**

1. Programming for problem solving and Data Structures

Course Objectives

- Explore various applications of computer graphics in fields such as design, simulation, animation, and visualization.
- Develop an understanding of 2D geometric transformations, viewing pipelines, and clipping algorithms for effective graphical representation.
- Understand the representation of 3D objects, including curves, surfaces, polygon rendering techniques, and various color models used in computer graphics.
- Learn and apply 3D geometric transformations and 3D clipping algorithms for modeling and viewing three-dimensional scenes.
- Analyze animation techniques and methods for visible surface detection to create realistic graphical scenes and sequences.

Course Outcomes

- Explore applications of computer graphics
- Understand 2D geometric transformations and clipping algorithms
- Understand 3D object representations, curves, surfaces, polygon rendering methods, color models
- Understand 3D geometric transformations and clipping algorithms
- Analyze animation sequence and visible surface detection methods

UNIT - I

Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random-scan systems, graphics monitors and work stations and input devices

Output primitives: Points and lines, line drawing algorithms (DDA and Bresenham's Algorithm) circle-generating algorithms and ellipse - generating algorithms

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms

UNIT - II

2-D geometric transformations: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, clipping operations, point clipping, Line clipping-Cohen Sutherland algorithms, Polygon clipping-Sutherland Hodgeman polygon clipping algorithm.

UNIT - III

3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, Polygon rendering methods, color models and color applications.

UNIT - IV

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

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

UNIT - V

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

Visible surface detection methods: Classification, back-face detection, depth-buffer method, BSP- tree method, area sub-division method and octree method.

TEXT BOOKS:

1. “Computer Graphics C version”, Donald Hearn and M. Pauline Baker, Pearson Education

REFERENCE BOOKS:

1. Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
2. Principles of Interactive Computer Graphics”, Neuman and Sproul, TMH.
3. Principles of Computer Graphics, Shalini Govil, Pai, 2005, Springer.
4. “Computer Graphics Principles & practice”, second edition in C, Foley, Van Dam, Feiner and Hughes, Pearson Education.
5. Computer Graphics, Steven Harrington, TMH.

CS 515 PE: EMBEDDED SYSTEMS (Professional Elective – II)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Pre-requisites:**

1. A course on “Digital Logic Design and Microprocessors”
2. A course on “Computer Organization and Architecture”

Course Objectives:

- Enable students to understand the criteria for selecting appropriate processors and microcontrollers for embedded applications.
- Provide knowledge of processor and microcontroller architectures, memory organization, real-world interfacing techniques, and interrupt handling mechanisms.
- Introduce the role and functionality of real-time operating systems (RTOS) in managing embedded system tasks.
- Develop the ability to design embedded firmware that meets functional and performance requirements.
- Foster analytical skills to evaluate the impact of task synchronization and latency on the performance and reliability of embedded systems.

Course Outcomes:

- Expected to understand the selection procedure of processors in the embedded domain.
- Understand the architecture of processors and microcontrollers, and demonstrate the ability to interface real-world devices and handle interrupts using appropriate service mechanisms.
- Expected to visualize the role of real-time operating systems in embedded systems.
- Design procedure of embedded firm ware.
- Expected to evaluate the correlation between task synchronization and latency issues

UNIT - I

Introduction to Embedded Systems: Processor embedded into a system, Embedded Hardware units and devices in a system, Embedded software in a system, Design process of an embedded system, classification of embedded systems, characteristics and quality attributes of an embedded systems

UNIT - II

Introduction to processor/microcontroller architecture, Real world interfacing, processor and memory organization, memory types, memory maps and addresses, interrupt sources and interrupt service mechanism.

UNIT - III

On board Communication Basics: serial; communication devices, Parallel devices, Wireless devices, Real time clock, Serial bus communication Protocols - I2C, SPI; Parallel buss communication - ISA, PCI.

UNIT - IV

Embedded Firmware Development: Overview of programming concepts - in assembly language and in high level language ‘C’, C Program elements- Heads, Source files, Processor Directives, Macros, Functions, Data types and Data Structures

UNIT - V

OS Based Embedded Systems: OS services - Process/Task Management, Memory Management, I/O subsystem manager, Inter Process/Task communications - Tasks, Task states, Shared data, Signals, Message Queues, Mailbox, Pipes and concepts of Semaphores.

TEXT BOOK:

1. Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill
2. Shibu K V, “Introduction to Embedded Systems”, Second Edition, Mc Graw Hill

REFERENCE BOOKS:

1. Rajkamal, Embedded Systems Architecture, Programming and Design, Tata McGraw-Hill
2. Frank Vahid and Tony Givargis, “Embedded Systems Design” - A Unified Hardware/Software Introduction, John Wiley

3. Lyla, “Embedded Systems” –Pearson
4. David E. Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint 2000.

CS 516 PE: INFORMATION RETRIEVAL SYSTEMS (Professional Elective – II)**B.Tech. III Year I Sem.**

L	T	P	C
3	0	0	3

Prerequisites:

1. Data Structures

Course Objectives:

- Enable students to understand the architecture, core components, and evaluation metrics of Information Retrieval systems.
- Train students in the use of indexing, ranking algorithms, and query processing techniques to efficiently retrieve relevant information from large-scale and unstructured datasets.
- Impart knowledge of key machine learning techniques used in document clustering and classification within the IR context.
- Guide students in implementing the essential components of a web search engine, including web crawling, indexing, document ranking, and relevance feedback mechanisms.
- Encourage exploration of advanced IR topics such as personalized search, multilingual retrieval, and performance evaluation using standard IR metrics and benchmarks.

Course Outcomes:

- To enable students to understand the architecture, components, and evaluation of Information Retrieval systems, and their role in web search and enterprise applications.
- To train students to use indexing, ranking, and query processing methods to retrieve relevant information from large-scale and unstructured data collections.
- To impart knowledge of document clustering and classification techniques
- To provide guidance to students through the implementation of core components of a web search engine including crawling, indexing, ranking, and relevance feedback mechanisms.
- To explore advanced topics in IR and evaluate system performance

UNIT - I

Introduction to Information Retrieval Systems: Definition of Information Retrieval System, Objectives of Information Retrieval Systems, Functional Overview, Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities. Difference between information and data retrieval systems, IRS Techniques – Retrieval Augmented generation, Challenges and Significance.

UNIT - II

Cataloging and Indexing: History and Objectives of Indexing, Components, features, models of Information retrieval systems, Information Extraction Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

UNIT - III

Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT - IV

User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Techniques and Tools.

UNIT - V

Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval-2D colors, Video Retrieval

TEXT BOOK:

1. Information Storage and Retrieval Systems – Theory and Implementation, Second Edition, Gerald J. Kowalski, Mark T. Maybury, Springer

REFERENCE BOOKS:

1. Frakes, W.B., Ricardo Baeza-Yates: Information Retrieval Data Structures and Algorithms, Prentice Hall, 1992.
2. Information Storage & Retrieval by Robert Korfhage – John Wiley & Sons.
3. Modern Information Retrieval by Yates and Neto Pearson Education.

CS 517 PE: DISTRIBUTED DATABASES (Professional Elective – II)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites:**

1. A course on “Database Management Systems”

Course Objectives:

- The purpose of the course is to enrich the previous knowledge of database systems and expose the need for distributed database technology to confront the deficiencies of the centralized database systems.
- Equip students with principles and knowledge of parallel and object-oriented databases.
- Develop the ability to analyze transaction properties in distributed systems and apply appropriate concurrency control and deadlock management strategies.
- Evaluate fault-tolerance and reliability measures in distributed database systems and explore parallel database architectures for efficient query processing and load balancing.
- Familiarize students with the structure, storage, and querying mechanisms of distributed object databases and provide insights into object-oriented data models, inheritance, object persistence, and comparisons between OODBMS and ORDBMS.

Course Outcomes:

- Understand theoretical, practical aspects and issues of distributed database systems.
- Understand the design aspects of object-oriented database systems and related developments.
- Analyze the properties and types of transactions in distributed systems, and apply concurrency control techniques, managing deadlocks efficiently.
- Evaluate reliability and fault-tolerance mechanisms in distributed DBMS and analyze parallel database architectures, query processing strategies, and load balancing techniques for optimized performance.
- Understand the architecture and components of distributed object databases and demonstrate knowledge of object-oriented data models, inheritance, persistence, and differences between OODBMS and ORDBMS.

UNIT - I

Introduction; Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas.

Distributed DBMS Architecture: Architectural Models for Distributed DBMS, DDMBS Architecture.

Distributed Database Design: Alternative Design Strategies, Distribution Design issues, Fragmentation, Allocation.

UNIT - II

Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.

Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

UNIT - III

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

UNIT - IV

Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning.

Parallel Database Systems: Parallel database system architectures, parallel data placement, parallel query

processing, load balancing, database clusters.

UNIT - V

Distributed object Database Management Systems: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing.

Object Oriented Data Model: Inheritance, object identity, persistent programming languages, persistence of objects, comparison OODBMS and ORDBMS

TEXT BOOKS:

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

REFERENCE BOOK:

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

CS 518 PE: NATURAL LANGUAGE PROCESSING (Professional Elective – II)**B.Tech. III Year I Sem.****L T P C**
3 0 0 3**Prerequisites:**

1. Data structures and compiler design

Course Objectives:

- Enable students to analyze the internal structure of words and documents using morphological and document structuring models, considering linguistic complexity and model performance.
- Equip students with the knowledge and skills to implement syntactic parsing techniques using data-driven approaches, such as Treebanks and parsing algorithms.
- Facilitate the resolution of syntactic ambiguities and introduce semantic interpretation techniques, including models of word sense, with attention to multilingual processing challenges.
- Develop proficiency in semantic parsing through the application of predicate-argument structures and various meaning representation frameworks.
- Train students to build, adapt, and evaluate diverse language models—such as N-gram, Bayesian, class-based, and topic-based—for both monolingual and multilingual natural language applications.

Course Outcomes:

- Analyze the structure of words and documents using morphological models and document structuring techniques, addressing linguistic complexity and model performance.
- Understand and implement syntactic parsing techniques using data-driven approaches like Treebanks and explore parsing algorithms for syntactic structure representation.
- Resolve syntactic ambiguities in multilingual contexts and perform semantic interpretation using models of word sense and system paradigms.
- Apply predicate-argument structures and meaning representation systems to enable deep semantic parsing of natural language texts.
- Build and evaluate various language models including N-gram, Bayesian, class-based, and topic-based models, and apply them in multilingual and cross-lingual NLP tasks.

UNIT - I**Finding the Structure of Words:** Words and Their Components, Issues and Challenges, Morphological Models**Finding the Structure of Documents:** Introduction, Methods, Complexity of the Approaches, Performances of the Approaches, Features**UNIT - II****Syntax I:** Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms**UNIT – III****Syntax II:** Models for Ambiguity Resolution in Parsing, Multilingual Issues**Semantic Parsing I:** Introduction, Semantic Interpretation, System Paradigms, Word Sense**UNIT - IV****Semantic Parsing II:** Predicate-Argument Structure, Meaning Representation Systems**UNIT - V****Language Modeling:** Introduction, N-Gram Models, Language Model Evaluation, Bayesian parameter estimation, Language Model Adaptation, Language Models- class based, variable length, Bayesian topic based, Multilingual and Cross Lingual Language Modeling

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.

REFERENCE BOOK:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

CS 504 PC: COMPUTER NETWORKS LAB**B.Tech. III Year I Sem.****L T P C****0 0 2 1****Course Objectives**

- To provide hands-on experience with data link layer framing methods
- To enable students to implement error detection and correction techniques
- To provide practical knowledge on routing algorithms and congestion control techniques
- To equip students to work with data representation formats and implement encoding and decoding techniques used in the presentation layer
- To familiarize students with standard networking tools and simulators

Course Outcomes

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

List of Experiments

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

How to run Nmap scan

Operating System Detection using Nmap 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 BOOK:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI.

REFERENCE BOOKS:

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

CS 505 PC: DEVOPS LAB**B.Tech. III Year I Sem.****L T P C**
0 0 2 1**Course Objectives:**

- To enable students to develop simple web applications through form design and validation
- To Provide hands-on experience in version control using Git and GitHub
- To Equip students with the ability to install, configure, and use Jenkins to automate the software build, test, and deployment processes through customized CI/CD pipelines.
- To impart practical knowledge of containerization using Docker and orchestration using Kubernetes
- To implement automated testing for web applications using Selenium

Course Outcomes:

- Able to write functional code for a simple user registration form, understanding the key concepts of form validation, user input handling, and data submission.
- Ability to gain practical experience with Git and GitHub commands, enabling them to manage and track changes in source code effectively
- Ability to install, configure, and use Jenkins to create continuous integration pipelines, facilitating automated building, testing, and deployment of applications
- Ability to explore Docker commands and learn to develop and manage containerized applications and integrate Docker with Kubernetes
- Develop and implement test cases for containerized applications using Selenium

List of Experiments:

1. Write code for a simple user registration form for an event.
2. Explore Git and GitHub commands.
3. Practice Source code management on GitHub. Experiment with the source code in exercise 1.
4. Jenkins installation and setup, explore the environment.
5. Demonstrate continuous integration and development using Jenkins.
6. Explore Docker commands for content management.
7. Develop a simple containerized application using Docker.
8. Integrate Kubernetes and Docker
9. Automate the process of running containerized application for exercise 7 using Kubernetes.
10. Install and Explore Selenium for automated testing.
11. Write a simple program in JavaScript and perform testing using Selenium.
12. Develop test cases for the above containerized application using selenium.

TEXT BOOKS:

1. Joakim Verona., Practical DevOps, Packt Publishing, 2016.

REFERENCE BOOKS:

1. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications.
2. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley.

EG 506 HS: ADVANCED ENGLISH COMMUNICATION SKILLS LAB**B.Tech. III Year I Sem.**

L	T	P	C
0	0	2	1

1. Introduction

The introduction of the Advanced English Communication Skills Lab is considered essential at the B.Tech 3rd year level. At this stage, the students need to prepare themselves for their career which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use appropriate English and perform the following:

1. Gathering ideas and information to organise ideas relevantly and coherently.
2. Making oral presentations.
3. Writing formal letters.
4. Transferring information from non-verbal to verbal texts and vice-versa.
5. Writing project/research reports/technical reports.
6. Participating in group discussions.
7. Engaging in debates.
8. Facing interviews.
9. Taking part in social and professional communication.

2. Course Objectives:

This Lab focuses on using multi-media instruction for language development to meet the following targets:

- To improve the students' fluency in English, with a focus on vocabulary
- To enable them to listen to English spoken at normal conversational speed by educated English speakers
- To respond appropriately in different socio-cultural and professional contexts
- To communicate their ideas relevantly and coherently in writing
- To prepare the students for placements.

3. Course Outcomes:

- Develop advanced listening and reading comprehension skills
- Ability to communicate their ideas relevantly and coherently in writing
- Ability to demonstrate proficiency in oral presentations with PPTs delivering well-structured presentations
- Able to effectively participate in group discussions, mastering the dynamics of GD through strategies such as voice modulation, idea organization, and intervention
- Ability to gain the skills necessary to prepare for and perform effectively in various types of interviews

4. Syllabus:

The following course content to conduct the activities is prescribed for the Advanced English Communication Skills (AECS) Lab:

1. **Activities on Listening and Reading Comprehension:** Active Listening – Development of Listening Skills Through Audio clips - Benefits of Reading – Methods and Techniques of Reading – Basic Steps to Effective Reading – Common Obstacles – Discourse Markers or Linkers - Sub- skills of reading - Reading for facts, negative facts and Specific Details- Guessing Meanings from Context, Inferring Meaning - Critical Reading — Reading Comprehension – Exercises for Practice.
2. **Activities on Writing Skills:** Vocabulary for Competitive Examinations - Planning for Writing – Improving Writing Skills - Structure and presentation of different types of writing – Free Writing and Structured Writing - Letter Writing –Writing a Letter of Application –Resume vs. Curriculum Vitae – Writing a Résumé – Styles of Résumé - e-Correspondence – Emails – Blog Writing - (N)etiquette – Report Writing – Importance of Reports – Types and Formats of Reports– Technical Report Writing– Exercises for Practice.
3. **Activities on Presentation Skills** - Starting a conversation – responding appropriately and relevantly – using the right language and body language – Role Play in different situations including Seeking

Clarification, Making a Request, Asking for and Refusing Permission, Participating in a Small Talk – Oral presentations (individual and group) through JAM sessions- PPTs – Importance of Presentation Skills – Planning, Preparing, Rehearsing and Making a Presentation – Dealing with Glossophobia or Stage Fear – Understanding Nuances of Delivery - Presentations through Posters/Projects/Reports – Checklist for Making a Presentation and Rubrics of Evaluation

4. **Activities on Group Discussion (GD):** Types of GD and GD as a part of a Selection Procedure - Dynamics of Group Discussion- Myths of GD - Intervention, Summarizing - Modulation of Voice, Body Language, Relevance, Fluency and Organization of Ideas – Do's and Don'ts - GD Strategies – Exercises for Practice.
5. **Interview Skills:** Concept and Process - Interview Preparation Techniques - Types of Interview Questions – Pre-interview Planning, Opening Strategies, Answering Strategies - Interview Through Tele-conference & Video-conference - Mock Interviews.

4. Minimum Requirement:

The Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab:

- Spacious room with appropriate acoustics
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- One PC with latest configuration for the teacher
- T. V, a digital stereo & Camcorder
- Headphones of High quality

5. **Suggested Software:** The software consisting of the prescribed topics elaborated above should be procured and used.

- **TOEFL & GRE** (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS)
- **Oxford Advanced Learner's Dictionary**, 10th Edition
- **Cambridge Advanced Learner's Dictionary**
- **DELTA's key to the Next Generation TOEFL Test: Advanced Skill Practice.**
- **Lingua TOEFL CBT Insider**, by Dreamtech

6. Books Recommended:

1. Rizvi, M. Ashraf (2018). *Effective Technical Communication*. (2nd ed.). McGraw Hill Education (India) Pvt. Ltd.
2. Suresh Kumar, E. (2015). *Engineering English*. Orient BlackSwan Pvt. Ltd.
3. Bailey, Stephen. (2018). *Academic Writing: A Handbook for International Students*. (5th Edition). Routledge.
4. Koneru, Aruna. (2016). *Professional Communication*. McGraw Hill Education (India) Pvt. Ltd.
5. Raman, Meenakshi & Sharma, Sangeeta. (2022). *Technical Communication, Principles and Practice*. (4th Edition) Oxford University Press.
6. Anderson, Paul V. (2007). *Technical Communication*. Cengage Learning Pvt. Ltd. New Delhi.
7. McCarthy, Michael; O'Dell, Felicity & Redman, Stuart. (2017). *English Vocabulary in Use* Series. Cambridge University Press
8. Sen, Leela. (2009). *Communication Skills*. PHI Learning Pvt Ltd., New Delhi.
9. Elbow, Peter. (1998). *Writing with Power*. Oxford University Press.
10. Goleman, Daniel. (2013). *Emotional Intelligence: Why it can matter more than IQ*. Bloomsbury Publishing.

CS 507 PC: UI DESIGN-FLUTTER**B.Tech. III Year I Sem.****L T P C****0 0 2 1****Prerequisites:**

1. Programming for problem-solving, Object-Oriented languages.

Course Objectives:

- To enable students to understand and implement core flutter widgets and layout design
- To teach students how to build responsive UI designs and implement navigation mechanisms
- To guide students in designing custom widgets and apply consistent styling across apps
- To provide practical skills in designing forms with input validation and error handling mechanisms
- To integrate data fetching, animations, and unit testing into Flutter applications

Course Outcomes:

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

List of Experiments: Students need to implement the following experiments

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

TEXT BOOK:

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

MC508: INTELLECTUAL PROPERTY RIGHTS**B.Tech. III Year I Sem.****L T P C**
3 0 0 0**Course Objectives:**

- To provide students with an understanding of the fundamentals and importance of Intellectual Property Rights (IPR) and their role in innovation, creativity, and economic development.
- To familiarize students with the trademark registration process and legal frameworks
- To equip students with the knowledge of developing the ability to analyze copyright ownership and international laws
- To provide an understanding of patent rights, ownership, and licensing
- To create awareness about intellectual property audits and management practices

Course Outcomes:

- Understand the fundamentals and significance of intellectual property.
- Demonstrate proficiency in the trademark registration process.
- Analyze copyright ownership issues and international copyright law.
- Understand the patent ownership rights and transfer.
- Understand the significance of intellectual property audits

UNIT – I

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, International copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

TEXT BOOK:

1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

REFERENCE BOOK:

1. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

CS 601 PC: MACHINE LEARNING**B.Tech. III Year II Sem.****L T P C**
3 0 0 3**Course Objectives:**

- To introduce students to the basic concepts and techniques of Machine Learning.
- To familiarize students with the architecture and application of Neural Networks in machine learning
- To enable students to analyze, design and apply machine learning techniques to real-world problems
- To provide knowledge of genetic algorithms and their role in optimization
- To introduce analytical learning approaches and domain-specific learning strategies

Course Outcomes:

- Understand the concepts of computational intelligence like machine learning
- Understand the Neural Networks and its usage in machine learning application.
- Ability to get the skill to apply machine learning techniques to address the real time problems in different areas
- Gain the skill to apply genetic algorithms to hypothesis space search problems.
- Gain knowledge in analytical learning techniques, such as explanation-based learning and learning with perfect domain theories.

UNIT - I

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept Learning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants: – Perceptron – Linear Separability – Linear Regression.

UNIT - II

Multi-layer Perceptron– Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

UNIT - III

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms

UNIT - IV

Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Least Squares Optimization – Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms

UNIT - V

Reinforcement Learning – Overview – Getting Lost Example
Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods

TEXT BOOKS:

1. Stephen Marsland, —Machine Learning – An Algorithmic Perspective, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

REFERENCE BOOKS:

1. Tom M Mitchell, —Machine Learning, First Edition, McGraw Hill Education, 2013.
2. Peter Flach, —Machine Learning: The Art and Science of Algorithms that Make Sense of Data, First Edition, Cambridge University Press, 2012.
3. Jason Bell, —Machine learning – Hands on for Developers and Technical Professionals, First Edition, Wiley, 2014
4. Ethem Alpaydin, —Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series), Third Edition, MIT Press, 2014

CS 602 PC: FORMAL LANGUAGES AND AUTOMATA THEORY**B.Tech. III Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- To provide students with foundational knowledge of abstract machines and their role in language recognition
- To enable the application of finite state machines for problem modeling and solving real-world scenarios
- To equip students to construct and analyze context-free grammars
- To develop an understanding of the theoretical limits of computation by identifying which problems can be solved algorithmically and which fall into the categories of undecidability or intractability.
- To build proficiency in mathematical tools and formal reasoning techniques

Course Outcomes

- Understand the concept of abstract machines and their power to recognize the languages.
- Employ finite state machines for modeling and solving computing problems.
- Design context free grammars for formal languages.
- Distinguish between decidability and undecidability.
- Gain proficiency with mathematical tools and formal methods

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with ϵ -transitions to NFA without ϵ -transitions. Conversion of NFA to DFA, Definition of Melay and Moore Machine, Conversion of Melay machine to Moore Machine, Conversion of Moore Machine to Melay machine

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV

Normal Forms for Context-Free Grammars: Eliminating useless symbols, Eliminating ϵ -Productions.

Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications

Closure Properties of Context-Free Languages: Closure properties of CFL's, Decision Properties of CFL's
Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT - V

Types of Turing machine: Turing machines and halting

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekar, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Textbook on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

CS 603 PC: ARTIFICIAL INTELLIGENCE**B.Tech. III Year II Sem.****L T P C****3 0 0 3****Prerequisites:**

1. Programming for problem solving, Data Structures.

Course Objectives:

- To equip students with the ability to translate real-world problems expressed in natural language into formal problem spaces and apply uninformed and informed search algorithms for solution finding.
- To provide understanding and hands-on experience with game-playing strategies and constraint satisfaction techniques using propositional logic for structured problem domains.
- To enable students to use appropriate knowledge representation and inference techniques
- To familiarize students with the design and use of ontologies for structured knowledge representation, and teach classical planning methods for automated reasoning and action sequencing in AI systems.
- To apply probabilistic reasoning for intelligent decision-making under uncertainty

Course Outcomes:

- Ability to formulate an efficient problem space for a problem expressed in natural language and apply suitable search algorithms.
- Apply adversarial search and constraint satisfaction techniques using propositional logic.
- Represent and infer knowledge using the appropriate technique for a given problem.
- Demonstrate understanding of knowledge representation using ontologies and develop classical planning solutions using AI techniques.
- Apply probabilistic reasoning for decision-making under uncertainty.

UNIT - I

Introduction to AI, Foundations of AI, History of AI, Applications of AI. Intelligent Agents, problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces

UNIT - II**Problem Solving by Search-II and Propositional Logic**

Adversarial Search: Games, Optimal Decisions in Games, Alpha-Beta Pruning, Imperfect Real-Time Decisions. **Constraint Satisfaction Problems:** Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems. **Propositional Logic:** Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III**Logic and Knowledge Representation**

First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Resolution.

UNIT - IV

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

Classical Planning: Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

UNIT - V

Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use

Probabilistic Reasoning: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

Introduction to Robot, Robotic Perception, application domains.

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

CS 608 PE: FULL STACK DEVELOPMENT (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C**
3 0 0 3**Pre-Requisites:**

1. Object Oriented Programming
2. Web Technologies

Course Objectives:

- Students will become familiar to implement fast, efficient, interactive and scalable web applications using run time environment provided by the full stack components.
- Apply Node.js packages to manage data, files, and handle HTTP requests and responses effectively in server-side development.
- Utilize MongoDB for scalable data storage and processing, and integrate it seamlessly with Node.js applications.
- Design and develop high-performance single-page applications (SPAs) using Express.js and Angular frameworks.
- Build interactive and reusable user interfaces using React components to enhance the client-side experience.

Course Outcomes:

- Understand Full stack components for developing web application.
- Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
- Use MongoDB data base for storing and processing huge data and connects with NodeJS application.
- Design faster and effective single page applications using Express and Angular.
- Create interactive user interfaces with react components.

UNIT-I**Introduction to Full Stack Development:**

Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks

UNIT-II**Node.js:**

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT-III**MongoDB:**

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV**Express and Angular:**

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT-V**React:**

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

CS 609 PE: INTERNET OF THINGS (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C****3 0 0 3****Pre-Requisites:** Computer organization, Computer Networks**Course Objectives:**

- To introduce the terminology, technology and its applications
- To introduce the concept of M2M (machine to machine) with necessary protocols
- To introduce the Python Scripting Language which is used in many IoT devices
- To introduce the Raspberry PI platform, that is widely used in IoT applications
- To introduce the implementation of web-based services on IoT devices

Course Outcomes:

- Interpret the impact and challenges posed by IoT networks leading to new architectural models.
- Compare and contrast the deployment of smart objects and the technologies to connect them to network.
- Develop IoT applications using Python by applying knowledge of data structures, control flow, file handling, and relevant Python libraries and packages essential for IoT systems.
- Interface physical IoT devices like Raspberry Pi using Python, and integrate with cloud platforms using RESTful APIs, WAMP, and frameworks like Django for real-time data exchange and management.
- Analyze and implement IoT-based solutions through real-world case studies such as home automation, environmental monitoring, and smart agriculture, showcasing the practical impact of IoT.

UNIT - I

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels and Deployment Templates

Domain Specific IoTs – Home automation, Environment, Agriculture, Health and Lifestyle

UNIT - II

IoT and M2M – M2M, Difference between IoT and M2M, SDN and NFV for IoT,

IoT System Management with NETCOZF, YANG- Need for IoT system Management, Simple Network management protocol, Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG

UNIT - III

IoT Systems – Logical design using Python-Introduction to Python – Python Data types & Data structures, Control flow, Functions, Modules, Packaging, File handling, Data/Time operations, Classes, Exception, Python packages of Interest for IoT

UNIT - IV

IoT Physical Devices and Endpoints - Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry PI with Python, Other IoT devices.

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs, WAMP-AutoBahn for IoT, Xively Cloud for IoT, Python web application framework

–Django, Designing a RESTful web API

UNIT V

Case studies- Home Automation, Environment-weather monitoring-weather reporting- air pollution monitoring, Agriculture.

TEXT BOOK:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madiseti, Universities Press, 2015, ISBN: 9788173719547.

REFERENCE BOOK:

1. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

CS 610 PE: SCRIPTING LANGUAGES (Professional Elective – III)**B.Tech. III 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 “Object Oriented Programming Concepts”.

Course Objectives:

- To introduce scripting languages and compare them with system and application programming languages
- To enable students to evaluate each scripting language based on syntax, flexibility, domain-specific applications, and ease of use to select the most appropriate language for a specific problem or task.
- To develop foundational programming skills in scripting languages
- To acquire advanced programming capabilities in Perl
- To explore the use of TCL and Tk for GUI and embedded systems development

Course Outcomes:

- Comprehend the differences between typical scripting languages and typical system and application programming languages
- Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.
- Acquire programming skills in scripting language
- Acquire advanced programming skills in Perl and understand its applications in various domains.
- Gain knowledge of the strengths and weaknesses of TCL and Tk and select an appropriate language for solving a given problem.

UNIT - I

Introduction: Ruby, Rails, The structure and Execution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and web services
RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System, Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

UNIT - III**Introduction to PERL and Scripting**

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT - IV**Advanced perl**

Finer points of looping, pack and unpack, filesystem, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT - V**TCL**

TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.

Tk

Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pragmatic Programmers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

CS 611 PE: MOBILE APPLICATION DEVELOPMENT (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C**
3 0 0 3**Prerequisites**

1. Acquaintance with JAVA programming
2. A Course on DBMS

Course Objectives

- To demonstrate their understanding of the fundamentals of Android operating systems
- To improve their skills of using Android software development tools
- To demonstrate their ability to develop software with reasonable complexity on mobile platform
- To demonstrate their ability to deploy software to mobile devices
- To demonstrate their ability to debug programs running on mobile devices

Course Outcomes

- Understand the working of Android OS Practically.
- Develop Android user interfaces
- Develop Android applications that use explicit and implicit intents, handle broadcasts using broadcast receivers, and implement user notifications through toasts and notification messages.
- Implement persistent data storage in Android applications using internal files and shared preferences to save, retrieve, and manage user data effectively.
- Design and manage local databases using SQLite and perform CRUD operations, and apply content providers to securely share and manipulate structured data across applications.

UNIT - I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes
Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring unit - s Layouts – Linear, Relative, Grid and Table Layouts
User Interface (UI) Components –Editable and non-editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers
Event Handling – Handling clicks or changes of various UI components
Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS
Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity
Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V

Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

TEXT BOOK:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012

REFERENCE BOOKS:

1. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
2. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

CS 612 PE: SOFTWARE TESTING METHODOLOGIES (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C**
3 0 0 3**Prerequisites**

1. Software Engineering

Course Objectives

- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- Enable students to understand and apply strategies in data flow testing and domain testing to detect software anomalies effectively.
- Develop logical and systematic approaches to testing through logic-based test strategies to enhance software verification.
- Introduce graph matrices and their applications to model and analyze control flow and improve test coverage.
- To develop skills in software test automation and management using the latest tools.

Course Outcomes:

- Understand purpose of testing and path testing
- Understand strategies in data flow testing and domain testing
- Develop logic-based test strategies
- Understand graph matrices and its applications
- Implement test cases using any testing automation tool

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs
 Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: transaction flows, transaction flow testing techniques.
 Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.
 Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III

Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
 Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.

4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

CS 618 OE: DATA STRUCTURES (Open Elective – I)**B.Tech. III Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites

1. A course on “Programming for Problem Solving

Course Objectives

- Provide students with the ability to implement and use linear data structures to solve computational problems effectively.
- Equip students with knowledge of dictionaries and hashing techniques for efficient data access and manipulation.
- Enable the design and manipulation of advanced search tree structures to improve the performance of search, insert, and delete operations.
- Introduce graph data structures and related algorithms for traversal and sorting, supporting efficient handling of complex relationships and large datasets.
- Apply string pattern matching techniques and trie-based structures for high-performance text processing and data retrieval applications.

Course Outcomes

- Implement and apply linear data structures such as arrays, linked lists, stacks, and perform operations like insertion, deletion, and searching.
- Apply various representations and operations on dictionaries and implement efficient hashing techniques with different collision resolution strategies including rehashing and extendible hashing.
- Construct and manipulate various search tree structures to optimize search, insertion, and deletion operations.
- Implement graph data structures using different methods, perform graph traversals, and apply sorting algorithms for both internal and external data.
- Apply string pattern matching algorithms such as Brute Force, Boyer-Moore, and KMP, and implement trie-based data structures for efficient string storage and retrieval operations.

UNIT - I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks- Operations, array and linked representations of stacks, stack applications, Queues- operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching. Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing- linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, B- Trees, B+ Trees, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.
Sorting: Quick Sort, Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

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

REFERENCE BOOK:

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

CS 619 OE: DATABASE MANAGEMENT SYSTEMS (Open Elective – I)**B.Tech. III Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites: A course on “Data Structures”.**Course Objectives:**

- Introduce the fundamental concepts of database systems and database design methodologies.
- Enable students to apply relational model concepts, enforce integrity constraints, formulate queries using relational algebra and calculus, and manage views and schema modifications.
- Develop proficiency in SQL for effective data retrieval, manipulation, and definition.
- Familiarize students with transaction management concepts, including concurrency control and recovery techniques.
- Provide an understanding of database storage architectures, indexing, and file organization for efficient data access and storage management.

Course Outcomes:

- Gain knowledge of fundamentals of DBMS and database design
- Apply the principles of the relational model by enforcing integrity constraints, designing relational schemas, formulating queries using relational algebra and relational calculus, and managing views and schema modifications effectively.
- Master the basics of SQL for retrieval and management of data.
- Be acquainted with the basics of transaction processing and concurrency control.
- Familiarity with database storage structures and access techniques

UNIT - I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II

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

UNIT - III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

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

UNIT - V

Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes,

Index data Structures, Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees: A Dynamic Index Structure.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 604 PC: MACHINE LEARNING LAB**B.Tech. III Year II Sem.**

L	T	P	C
0	0	2	1

Course Objective: The objective of this lab is to get an overview of the various machine learning techniques and can demonstrate them using python.

- To provide students with an understanding of the foundational concepts and complexities of Machine Learning algorithms
- To equip students with knowledge of modern data analysis frameworks and paradigms
- To enable practical implementation of standard Machine Learning algorithms
- To Provide students with hands-on experience to perform experimental analysis using real-world datasets
- To promote critical understanding of algorithm applicability and performance trade-offs

Course Outcomes:

- Understand complexity of Machine Learning algorithms and their limitations
- Understand modern notions in data analysis-oriented computing
- Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own
- Be capable of performing experiments in Machine Learning using real-world data.
- Understand the complexity of Machine Learning algorithms and their limitations.

List of Experiments

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode
Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

CS 605 PC: ARTIFICIAL INTELLIGENCE LAB**B.Tech. III Year II Sem.**

L	T	P	C
0	0	2	1

Course Objectives:

- To provide a foundation in solving classical AI problems using uninformed search techniques
- To equip students with the ability to define problem states, actions, and goal conditions, and apply state transition representations to model and solve real-world problem-solving scenarios.
- To introduce optimization and heuristic search techniques for AI applications
- To provide students with skills in designing adversarial agents using game trees
- To promote the use of logical reasoning and goal-driven problem-solving approaches

Course Outcomes:

- Implement uninformed search algorithms to solve classical AI problems.
- Apply AI techniques to solve classical problem-solving scenarios using state transition logic.
- Analyze and solve optimization and heuristic-based problems using informed search and approximation methods.
- Design and develop adversarial game-playing agents using game trees and pruning strategies.
- Demonstrate the application of logic, rules, and goal-based reasoning in classical AI problems.

LIST OF EXPERIMENTS

Write a Program to Implement the following using Python.

1. Breadth First Search
2. Depth First Search
3. Tic-Tac-Toe game
4. 8-Puzzle problem
5. Water-Jug problem
6. Travelling Salesman Problem
7. Tower of Hanoi
8. Monkey Banana Problem
9. Alpha-Beta Pruning
10. 8-Queens Problem

TEXT BOOK:

1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3. Artificial Intelligence, Shivani Goel, Pearson Education.

CS 613 PE: FULL STACK DEVELOPMENT LAB (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C**
0 0 2 1**Pre-Requisites:**

1. Object Oriented Programming
2. Web Technologies

Course Objectives:

- Enable students to build flexible and interactive web applications using front-end frameworks such as Angular and React, and server-side platforms like Node.js and Express.
- Train students to perform efficient CRUD operations and manage large volumes of data using MongoDB as the backend database.
- Facilitate the development of real-time and component-driven web interfaces using React components and state management techniques.
- Develop proficiency in handling HTTP requests, responses, file operations, and user interactions through full stack module integration.
- Guide students in building end-to-end web applications by connecting front-end and back-end components, implementing user authentication, routing, and data flow using modular and maintainable code structures.

Course Outcomes:

- Design flexible and responsive Web applications using Node JS, React, Express and Angular.
- Perform CRUD operations with MongoDB on huge amount of data.
- Develop real time applications using react components.
- Use various full stack modules to handle http requests and responses.
- Develop end-to-end web applications by integrating front-end frameworks (Angular/React) with back-end technologies (Node.js/Express) and perform file handling, user authentication, routing, and state management using modular full-stack components.

List of Experiments

1. Create an application to setup node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response using NodeJS
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Develop a Job Registration form and validate it using angular JS.
11. Write an angular JS application to access JSON file data of an employee from a server using \$http service.
12. Develop a web application to manage student information using Express and Angular JS.
13. Write a program to create a simple calculator Application using React JS.
14. Write a program to create a voting application using React JS
15. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.
16. Build a music store application using react components and provide routing among the web pages.
17. Create a react application for an online store which consist of registration, login, product information pages and implement routing to navigate through these pages.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.

2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Brad Green & Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.
4. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.

CS 614 PE: INTERNET OF THINGS LAB (PROFESSIONAL ELECTIVE – III)**B.Tech. III Year II Sem.**

L	T	P	C
0	0	2	1

Course Objectives

- Provide hands-on experience in programming sensors and actuators using embedded platforms like Raspberry Pi, Arduino, and NodeMCU for real-time IoT applications.
- Enable students to install and configure operating systems on Raspberry Pi and manage boot processes across different environments.
- Develop proficiency in using Python and GPIO libraries to interact with hardware interfaces and implement PWM-based control mechanisms.
- Train students to build web applications using Django by creating views, templates, and RESTful API integrations for IoT-based data visualization and interaction.
- Facilitate integration of Django applications with MySQL databases to store, manage, and retrieve sensor and user data efficiently in a full-stack IoT environment.

Course Outcomes

- Ability to interface and program sensors (distance, temperature, LDR) and actuators (LEDs, switches) using embedded hardware platforms like Raspberry Pi, Arduino, and NodeMCU for IoT applications.
- Set up and configure Raspberry Pi by installing operating systems using various tools, and manage the booting process.
- Use Python and GPIO libraries to access and control hardware peripherals such as LEDs through GPIO pins, including PWM for LED brightness control.
- Design and develop Django-based web applications by creating projects, apps, templates, and views that connect to external REST APIs like weather stations.
- Integrate Django applications with MySQL to enable backend data storage and retrieval for full-stack IoT web systems.

List of Experiments

1. Using Raspberry pi
 - a. Calculate the distance using a distance sensor.
 - b. Interface an LED and switch with Raspberry pi.
 - c. Interface an LDR with Raspberry Pi.
2. Using Arduino
 - a. Calculate the distance using a distance sensor.
 - b. Interface an LED and switch with Aurdino.
 - c. Interface an LDR with Aurdino
 - d. Calculate temperature using a temperature sensor.
3. Using Node MCU
 - a. Calculate the distance using a distance sensor.
 - b. Interface an LED and switch with Raspberrypi.
 - c. Interface an LDR with Node MCU
 - d. Calculate temperature using a temperature sensor.
4. Installing OS on Raspberry Pi
 - a) Installation using PiImager
 - b) Installation using image file
 - Downloading an Image
 - Writing the image to an SD card
 - using Linux
 - using Windows
 - Booting up Follow the instructions given in the URL_

<https://www.raspberrypi.com/documentation/computers/getting-started.html>

5. Accessing GPIO pins using Python
 - a) Installing GPIO Zero library.
update your repositories list:
install the package for Python 3:
 - b) Blinking an LED connected to one of the GPIO pin
 - c) Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.
6. Create a DJANGO project and an app.
7. Create a DJANGO view for weather station REST API
8. Create DJANGO template
9. Configure MYSQL with DJANGO framework

TEXT BOOKS:

1. Internet of Things - A Hands-on Approach, Arshdeep Bahga and Vijay Madisetti, Universities Press, 2015, ISBN: 9788173719547.
2. Getting Started with Raspberry Pi, Matt Richardson & Shawn Wallace, O'Reilly (SPD), 2014, ISBN: 9789350239759.

REFERENCE BOOKS:

1. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer, 2016
2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

CS615PE: SCRIPTING LANGUAGES LAB (Professional Elective – III)**B.Tech. III Year II Sem.**

L	T	P	C
0	0	2	1

Prerequisites: Any High level programming language (C, C++)**Course Objectives**

- To introduce the fundamental differences between scripting and programming languages
- To provide students with hands-on experience in writing Ruby scripts
- To train students to use Perl for real-world applications involving string manipulation, regular expressions and CGI scripting
- To enable students to develop skills in developing TCL scripts
- To enhance the ability to choose the appropriate scripting language for specific tasks

Course Outcomes

- Understand the differences between Scripting languages and programming languages
- Gain some fluency programming in Ruby
- Gain some fluency programming in Perl
- Gain some fluency programming in TCL
- Understand the differences between scripting languages and programming languages.

LIST OF EXPERIMENTS

1. Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer
2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.
3. Write a Ruby script which accept the users first and last name and print them in reverse order with a space between them
4. Write a Ruby script to accept a filename from the user print the extension of that
5. Write a Ruby script to find the greatest of three numbers
6. Write a Ruby script to print odd numbers from 10 to 1
7. Write a Ruby script to check two integers and return true if one of them is 20 otherwise return their sum
8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100
9. Write a Ruby script to print the elements of a given array
10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash
11. Write a TCL script to find the factorial of a number
12. Write a TCL script that multiplies the numbers from 1 to 10
13. Write a TCL script for sorting a list using a comparison function
14. Write a TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv) Concatenate the list
15. Write a TCL script to comparing the file modified times.
16. Write a TCL script to Copy a file and translate to native format.
17.
 - a) Write a Perl script to find the largest number among three numbers.
 - b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.
18. Write a Perl program to implement the following list of manipulating functions
 - a) Shift
 - b) Unshift
 - c) Push
19.
 - a) Write a Perl script to substitute a word, with another word in a string.
 - b) Write a Perl script to validate IP address and email address.
20. Write a Perl script to print the file in reverse order using command line arguments

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.

2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pragmatic Programmers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

CS 616 PE: MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective – III)**B.Tech. III Year II Sem.**

L	T	P	C
0	0	2	1

Prerequisites: --- NIL---**Course Objectives:**

- Enable students to design and develop basic Android applications using various UI components and layout structures for effective user interaction.
- Provide the ability to implement modular and responsive interfaces using fragments, lists, and event-driven programming, including support for device orientation and adaptive navigation.
- Train students to integrate Android system capabilities using intents, notifications, and toast messages to perform communication and user feedback operations.
- Develop proficiency in persistent data management using text files, Shared Preferences, and SQLite databases for user authentication and information handling.
- Familiarize students with Android device features and services such as contact access, alarm scheduling, and background processing to create real-world, context-aware mobile applications.

Course Outcomes:

- Design and develop basic Android applications using UI components for structured form-based input and interaction.
- Implement dynamic navigation and data display using fragments, list views, and event listeners, including orientation-based UI adjustments and fragment transactions.
- Utilize Android Intents and Notifications to perform system-level actions such as dialing, messaging, opening websites, and displaying user notifications along with toast messages.
- Handle user data storage and authentication using text files, Shared Preferences, and SQLite databases to create login systems, registration forms, and admin panels with data persistence.
- Access and integrate Android device features and services, including contacts, alarm scheduling, and background services, to create functional and context-aware mobile applications.

LIST OF EXPERIMENTS:

1. (a) Create an Android application that shows Hello + name of the user and run it on an emulator.
(b) Create an application that takes the name from a text box and shows hello message along with the name entered in the text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Datepicker), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store usernames and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with a Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin can select any

record for edit or modify. The results should be reflected in the table.

10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.
12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

TEXT BOOKS:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers, James C Sheusi, Cengage, 2013.

REFERENCE BOOK:

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

CS 617 PE: SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective – III)**B.Tech. III Year II Sem.****L T P C**
0 0 2 1**Prerequisites**

1. A basic knowledge of programming.

Course Objectives

- Enable students to formulate optimal test strategies aligned with various software development models.
- Train students to create and implement graphical user interface (GUI), bitmap, and database checkpoints to verify application correctness and visual consistency.
- Develop proficiency in setting up and validating various types of database checkpoints including default, custom, and runtime checks to ensure data integrity.
- Provide hands-on experience in executing batch testing processes with and without parameter passing to automate repetitive test cases effectively.
- Equip students to perform advanced data-driven testing using flat files, grids, and Excel sheets, and execute tests in silent mode to assess application performance under diverse input scenarios.

Course Outcomes

- Design and develop the best test strategies in accordance with the development model.
- Design and develop GUI, Bitmap and database checkpoints
- Develop database checkpoints for different checks
- Perform batch testing with and without parameter passing
- Demonstrate the ability to design, automate, and execute data-driven and batch testing to validate application functionality under varying input conditions.

List of Experiments

1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
5. Database checkpoint for Default check
6. Database checkpoint for custom check
7. Database checkpoint for runtime record check
8. Data driven batch
9. Silent mode test execution without any interruption
10. Test case for calculator in windows application

TEXT BOOKS

1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Tools, Dr. K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS

1. The craft of software testing, Brian Marick, Pearson Education.

2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

CS 606 PW: BIG DATA-SPARK**B.Tech. III Year II Sem.****L T P C**
0 0 4 2

Course Objectives: The main objective of the course is to process Big Data with advance architecture like spark and streaming data in Spark.

- Enable students to develop and execute MapReduce programs for large-scale data analysis using Hadoop.
- Train students to write HiveQL queries to explore structured data and understand the components of the Spark ecosystem.
- Familiarize students with core RDD operations in Spark, including transformations such as filter, map, flatMap, and actions like count and distinct.
- Build competence in querying structured data using Spark SQL for efficient data manipulation and analysis.
- Apply Spark join operations on large datasets and utilize tools like **Sqoop** for data import/export between Hadoop and relational databases, enabling seamless data integration.

Course Outcomes:

- Develop MapReduce Programs to analyze large dataset Using Hadoop and Spark
- Write Hive queries to analyze large dataset Outline the Spark Ecosystem and its components
- Perform the filter, count, distinct, map, flatMap RDD Operations in Spark.
- Build Queries using Spark SQL
- Apply Spark joins on Sample Data Sets, making use of sqoop to import and export data from hadoop to database and vice-versa

List of Experiments:

1. To Study of Big Data Analytics and Hadoop Architecture
 - (i) know the concept of big data architecture
 - (ii) know the concept of Hadoop architecture
2. Loading DataSet in to HDFS for Spark Analysis
 - Installation of Hadoop and cluster management
 - (i) Installing Hadoop single node cluster in ubuntu environment
 - (ii) Knowing the differencing between single node clusters and multi-node clusters
 - (iii) Accessing WEB-UI and the port number
 - (iv) Installing and accessing the environments such as hive and sqoop
3. File management tasks & Basic linux commands
 - (i) Creating a directory in HDFS
 - (ii) Moving forth and back to directories
 - (iii) Listing directory contents
 - (iv) Uploading and downloading a file in HDFS
 - (v) Checking the contents of the file
 - (vi) Copying and moving files
 - (vii) Copying and moving files between local to HDFS environment
 - (viii) Removing files and paths
 - (ix) Displaying few lines of a file
 - (x) Display the aggregate length of a file
 - (xi) Checking the permissions of a file
 - (xii) Zipping and unzipping the files with & without permission pasting it to a location
 - (xiii) Copy, Paste commands
4. Map-reducing
 - (i) Definition of Map-reduce
 - (ii) Its stages and terminologies
 - (iii) Word-count program to understand map-reduce (Mapper phase, Reducer phase, Driver code)

5. Implementing Matrix-Multiplication with Hadoop Map-reduce
6. Compute Average Salary and Total Salary by Gender for an Enterprise.
7.
 - (i) Creating hive tables (External and internal)
 - (ii) Loading data to external hive tables from sql tables(or)Structured c.s.v using scoop
 - (iii) Performing operations like filterations and updations
 - (iv) Performing Join (inner, outer etc)
 - (v) Writing User defined function on hive tables
8. Create a sql table of employees Employee table with id,designation Salary table (salary ,dept id)
Create external table in hive with similar schema of above tables,Move data to hive using scoop and load the contents into tables,filter a new table and write a UDF to encrypt the table with AES-algorithm, Decrypt it with key to show contents
9.
 - (i) Pyspark Definition(Apache Pyspark) and difference between Pyspark, Scala, pandas
 - (ii) Pyspark files and class methods
 - (iii) get(file name)
 - (iv) get root directory()
10. Pyspark -RDD'S
 - (i) what is RDD's?
 - (ii) ways to Create RDD
 - (iii) parallelized collections
 - (iv) external dataset
 - (v) existing RDD's
 - (vi) Spark RDD's operations (Count, foreach(), Collect, join,Cache())
11. Perform pyspark transformations
 - (i) map and flatMap
 - (ii) to remove the words, which are not necessary to analyze this text.
 - (iii) groupBy
 - (iv) What if we want to calculate how many times each word is coming in corpus?
 - (v) How do I perform a task (say count the words 'spark' and 'apache' in rdd3) separatly on each partition and get the output of the task performed in these partition?
 - (vi) unions of RDD
 - (vii) join two pairs of RDD Based upon their key
12. Pyspark sparkconf-Attributes and applications
 - (i) What is Pyspark spark conf ()
 - (ii) Using spark conf create a spark session to write a dataframe to read details in a c.s.v and later move that c.s.v to another location

TEXT BOOKS:

1. Spark in Action, Marko Bonaci and Petar Zecevic, Manning.
2. PySpark SQL Recipes: With HiveQL, Dataframe and Graphframes, Raju Kumar Mishra and Sundar Rajan Raman, Apress Media.

MC607: ENVIRONMENTAL SCIENCE**B.Tech. III Year II Sem.**

L	T	P	C
3	0	0	0

Course Objectives:

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures.
- To instil the importance of biodiversity and conservation strategies
- To provide an understanding of environmental pollution and control technologies
- Understanding the environmental policies and regulations

Course Outcomes:

- Understand the classification, structure, function, energy flow, and biogeochemical cycles of ecosystems, including their value, services, and carrying capacity.
- Acquire knowledge about the classification of natural resources, understand the importance of water resources, and gain insights into the use, exploitation, and environmental effects of mineral resources.
- Recognize the value of biodiversity and gain knowledge about threats to biodiversity and conservation methods
- Understand the classification of environmental pollution and acquire knowledge about solid waste management and pollution control technologies
- Understand various environmental policies and legislation, and recognize the importance of environmental education, sustainable development, human health, and low carbon lifestyle towards a sustainable future.

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, Field visits.

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 resources:** Forest resources, **Energy resources:** growing energy needs, renewable and non-renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity and Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. 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. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **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:** Wastewater Treatment methods: Primary, secondary and Tertiary.

Overview of air pollution control technologies, Concepts of bioremediation. **Global Environmental Issues and Global Efforts:** Climate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit,

Kyoto protocol, and Montréal Protocol. NAPCC-GoI Initiatives.

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 Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission.
- 2 Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

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

CS 701 PC: CRYPTOGRAPHY AND NETWORK SECURITY**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course Objectives:**

- Understand various cryptographic algorithms.
- Train students to identify and analyze the security requirements of client-server-based information systems.
- Provide insight into contemporary legal, ethical, and policy issues related to information and cybersecurity.
- Develop a comprehensive understanding of transport-level security protocols (such as SSL/TLS) and wireless network security mechanisms.
- Familiarize students with advanced topics in cryptography and real-world security case studies to strengthen their ability to evaluate and apply security solutions.

Course Outcomes:

- Student will be able to understand basic cryptographic algorithms, message and web authentication and security issues.
- Ability to identify information system requirements for both of them such as client and server.
- Ability to understand the current legal issues towards information security.
- Develop an understanding of transport-level security protocols as well as wireless network security principles and protocols.
- Recognize advanced topics in cryptography and security, and various case studies

UNIT - I

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security **Cryptography**

Concepts and Techniques: Introduction, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks.

UNIT - II

Symmetric key Ciphers: Block Cipher principles, DES, AES, Blowfish, RC5, IDEA, Block cipher operation, Stream ciphers, RC4.

Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm.

UNIT - III

Cryptographic Hash Functions: Message Authentication, Secure Hash Algorithm (SHA-512), **Message authentication codes:** Authentication requirements, HMAC, CMAC, Digital signatures, Elgamal Digital Signature Scheme.

Key Management and Distribution: Symmetric Key Distribution Using Symmetric & Asymmetric Encryption, Distribution of Public Keys, Kerberos, X.509 Authentication Service, Public – Key Infrastructure

UNIT - IV

Transport-level Security: Web security considerations, Secure Socket Layer and Transport Layer Security, HTTPS, Secure Shell (SSH)

Wireless Network Security: Wireless Security, Mobile Device Security, IEEE 802.11 Wireless LAN, IEEE 802.11i Wireless LAN Security

UNIT - V

E-Mail Security: Pretty Good Privacy, S/MIME IP Security: IP Security overview, IP Security architecture, Authentication Header, Encapsulating security payload, Combining security associations, Internet Key Exchange

Case Studies on Cryptography and security: Secure Multiparty Calculation, Virtual Elections, Single sign On,

Secure Inter-branch Payment Transactions, Cross site Scripting Vulnerability.

TEXT BOOKS:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition
2. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition

REFERENCE BOOKS:

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

CS 702 PC: COMPILER DESIGN**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Prerequisites**

1. A course on “Formal Languages and Automata Theory”.
2. A course on “Computer Organization and architecture”.
3. A course on “Data Structures”.

Course Objectives:

- Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
- To equip students with the ability to design lexical analyzers using regular expressions and deterministic/non-deterministic finite automata, and implement them using tools like Lex.
- To train students to design context-free grammars and implement top-down (LL) and bottom-up (LR) parsing algorithms using parser generators
- Enable students to implement syntax-directed definitions and translation schemes for generating intermediate code.
- Introduce the concepts of activation records, symbol tables, heap and stack memory management, and enable students to design basic code generators and optimize generated code using peephole techniques and control-flow graphs.
- Learn about machine-independent code optimization techniques

Course Outcomes:

- Understand the structure and phases of a compiler, and perform lexical analysis using regular expressions, finite automata, and Lex tool.
- Design and implement context-free grammars, LL and LR parsers for syntax analysis, and use parser generators.
- Apply syntax-directed translation techniques and generate intermediate code including syntax trees, three-address code, and perform type checking.
- Analyze run-time environments, implement memory management strategies, and design simple code generators using flow graphs and peephole optimization.
- Apply machine-independent optimization techniques such as data-flow analysis, constant propagation, and loop optimizations to enhance code performance.

UNIT - I**Introduction:** The structure of a compiler, the science of building a compiler, programming language basics**Lexical Analysis:** The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.**UNIT - II****Syntax Analysis:** Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers, Using Ambiguous Grammars and Parser Generators.**UNIT - III****Syntax-Directed Translation:** Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.**Intermediate-Code Generation:** Variants of Syntax Trees, Three-Address Code, Types and Declarations, Type Checking, Control Flow, Switch-Statements, Intermediate Code for Procedures.**UNIT - IV****Run-Time Environments:** Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to Trace-Based Collection.**Code Generation:** Issues in the Design of a Code Generator, The Target Language, Addresses in the Target

Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, A Simple Code Generator, Peephole Optimization, Register Allocation and Assignment, Dynamic Programming Code-Generation

UNIT - V

Machine-Independent Optimization: The Principal Sources of Optimization, Introduction to Data-Flow Analysis, Foundations of Data-Flow Analysis, Constant Propagation, Partial-Redundancy Elimination, Loops in Flow Graphs.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Loudon, Thomson.

CS 706 PE: GRAPH THEORY (Professional Elective – IV)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course Objectives:**

- To introduce students to the foundational principles of graph theory, including the classification, representation, and construction of graphs, enabling them to model and analyze complex network-based problems.
- To develop the ability to understand graph connectivity and apply shortest path algorithms, preparing students to solve practical problems in routing, network design, and communication systems.
- To impart knowledge on the structural characteristics of trees and special classes of graphs, with a focus on algorithms for constructing minimum spanning trees and understanding Eulerian and Hamiltonian paths.
- To familiarize students with advanced graph concepts such as independent sets, coverings, and matchings, and to equip them with mathematical tools and algorithms for solving optimization problems in graph theory.
- To cultivate the skills necessary to apply vertex and edge coloring techniques in graphs, and to analyze their use in real-world scenarios like scheduling, frequency assignment, and resource allocation.

Course Outcomes:

- Understand the fundamental concepts of graph theory and apply these to problems such as graphic sequences and network modeling
- Apply the concepts of connected graphs, cut-vertices, blocks, and shortest paths in weighted graphs to solve real-world routing and connectivity problems.
- Analyze the properties of trees, spanning trees, and special graph.
- Evaluate independent sets, vertex covers, and matchings in bipartite graphs using theorems and approximation algorithms.
- Implement and analyze vertex and edge coloring algorithms and apply coloring concepts to scheduling and optimization problems.

UNIT - I

Introduction-Discovery of graphs, Definitions, Subgraphs, Isomorphic graphs, Matrix representations of graphs, Degree of a vertex, Directed walks, paths and cycles, Connectivity in digraphs, Eulerian and Hamilton digraphs, Eulerian digraphs, Hamilton digraphs, Special graphs, Complements, Larger graphs from smaller graphs, Union, Sum, Cartesian Product, Composition, Graphic sequences, Graph theoretic model of the LAN problem, Havel-Hakimi criterion, Realization of a graphic sequence.

UNIT - II

Connected graphs and shortest paths - Walks, trails, paths, cycles, Connected graphs, Distance, Cut-vertices and cut-edges, Blocks, Connectivity, Weighted graphs and shortest paths, Weighted graphs, Dijkstra's shortest path algorithm, Floyd-Warshall shortest path algorithm.

UNIT - III

Trees- Definitions and characterizations, Number of trees, Cayley's formula, Kirchoff's-matrix-tree theorem, Minimum spanning trees, Kruskal's algorithm, Prim's algorithm, Special classes of graphs, Bipartite Graphs, Line Graphs, Chordal Graphs, Eulerian Graphs, Fleury's algorithm, Chinese Postman problem, Hamilton Graphs, Introduction, Necessary conditions and sufficient conditions.

UNIT - IV

Independent sets coverings and matchings– Introduction, Independent sets and coverings: basic equations, Matchings in bipartite graphs, Hall's Theorem, Konig's Theorem, Perfect matchings in graphs, Greedy and approximation algorithms.

UNIT - V

Vertex Colorings- Basic definitions, Cliques and chromatic number, Mycielski's theorem, Greedy coloring algorithm, Coloring of chordal graphs, Brooks theorem, Edge Colorings, Introduction and Basics, Gupta-Vizing theorem, Class-1 and Class-2 graphs, Edge-coloring of bipartite graphs, Class-2 graphs, Hajos union and Class-2 graphs, A scheduling problem and equitable edge-coloring.

TEXT BOOKS:

1. J. A. Bondy and U. S. R. Murty. Graph Theory, volume 244 of Graduate Texts in Mathematics. Springer, 1st edition, 2008.
2. J. A. Bondy and U. S. R. Murty. Graph Theory with Applications.

REFERENCE BOOKS:

1. Lecture Videos: <http://nptel.ac.in/courses/111106050/13>
2. Introduction to Graph Theory, Douglas B. West, Pearson.

CS 707 PE: CYBER SECURITY (Professional Elective – IV)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course objectives:**

- To understand various types of cyber-attacks and cyber-crimes.
- To have an overview of the cyber laws & concepts of cyber forensics.
- To understand the evolving landscape of cybercrimes related to mobile and wireless devices, including common attack types and mobility trends.
- To analyze mobile security threats and evaluate authentication mechanisms and organizational security policies.
- To design effective security architectures and strategies tailored for mobile computing environments within organizations.
- To study the defensive techniques against these attacks.

Course Outcomes:

- Analyze and evaluate the cyber security needs of an organization.
- Understand Cyber Security Regulations and Roles of International Law.
- Understand the nature of cybercrimes associated with mobile and wireless devices, analyze security threats and evaluate organizational policies and authentication mechanisms in mobile computing environments.
- Design and develop security architecture for an organization.
- Understand fundamental concepts of data privacy attacks.

UNIT- I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy. Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics.

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

UNIT- IV

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations.

UNIT - V

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc.

Cybercrime: Examples and Mini-Cases

Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances.

Mini-Cases: The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

TEXT BOOKS:

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley

REFERENCE BOOKS:

1. B. B. Gupta, D.P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press
2. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
3. Introduction to Cyber Security, Chwan-Hwa(john) Wu, J.David Irwin, CRC Press T&F Group.

CS 708 PE: SOFT COMPUTING (Professional Elective – IV)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course Objectives:**

- Familiarize with soft computing concepts
- Introduce and use the idea of fuzzy logic and use of heuristics based on human experience
- Familiarize the Neuro-Fuzzy modeling using Classification and Clustering techniques
- Learn the concepts of Genetic algorithm and its applications
- Acquire the knowledge of Rough Sets.

Course Outcomes:

- Identify the difference between Conventional Artificial Intelligence to Computational Intelligence.
- Understand fuzzy logic and reasoning to handle and solve engineering problems
- Apply fuzzy logic principles and decision-making techniques to model uncertainty in real-world problems, and to solve complex, multi-dimensional optimization tasks efficiently.
- Apply the Classification techniques on various applications.
- Perform various operations of genetic algorithms and Rough Sets.

UNIT - I

Introduction to Soft Computing: Evolutionary Computing, "Soft" computing versus "Hard" computing, Soft Computing Methods, Recent Trends in Soft Computing, Characteristics of Soft computing, Applications of Soft Computing Techniques.

UNIT- II

Fuzzy Systems: Fuzzy Sets, Fuzzy Relations, Fuzzy Logic, Fuzzy Rule-Based Systems

UNIT- III

Fuzzy Decision Making, Particle Swarm Optimization

UNIT- IV

Genetic Algorithms: Basic Concepts, Basic Operators for Genetic Algorithms, Crossover and Mutation Properties, Genetic Algorithm Cycle, Fitness Function, Applications of Genetic Algorithm.

UNIT- V

Rough Sets, Rule Induction, and Discernibility Matrix, Integration of Soft Computing Techniques.

TEXT BOOK:

1. Soft Computing – Advances and Applications - Jan 2015 by B.K. Tripathy and J. Anuradha – Cengage Learning

REFERENCE BOOKS:

1. S. N. Sivanandam & S. N. Deepa, "Principles of Soft Computing", 2nd edition, Wiley India, 2008.
2. David E. Goldberg, "Genetic Algorithms-In Search, optimization and Machine learning", Pearson Education.
3. J. S. R. Jang, C.T. Sun and E.Mizutani, "Neuro-Fuzzy and Soft Computing", Pearson Education, 2004.
4. G.J. Klir & B. Yuan, "Fuzzy Sets & Fuzzy Logic", PHI, 1995.
5. Melanie Mitchell, "An Introduction to Genetic Algorithm", PHI, 1998.
6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", McGraw- Hill International editions, 1995

CS 709 PE: CLOUD COMPUTING (Professional Elective – IV)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Pre-requisites:**

1. A course on “Computer Networks”.
2. A course on “Operating System”.

Course Objectives:

- Introduce students to traditional and modern computing models, highlighting their characteristics and use cases.
- Provide students with knowledge and hands-on skills to develop, deploy, and manage applications on cloud platforms using different programming environments and cloud-native tools.
- Familiarize students with the essential components of cloud architecture and the layered service delivery models.
- Enable students to critically analyze and compare public, private, hybrid, and community cloud deployment models in terms of scalability, cost-efficiency, security, and performance.
- To gain awareness of leading cloud service providers and their offerings

Course Outcomes:

- Understand and differentiate between various computing paradigms
- Ability to understand the ways in which the cloud can be programmed and deployed
- Ability to understand various service delivery models of a cloud computing architecture
- Understand and evaluate the different cloud service models
- Understanding cloud service providers

UNIT - I

Computing Paradigms, Cloud Computing Fundamentals, Cloud Computing Architecture and Management

UNIT - II**Cloud Deployment Models, Cloud Service Models, Technological Drivers for Cloud Computing:** SOA and Cloud, Multicore Technology, Web 2.0 and Web 3.0, Pervasive Computing, Operating System, Application Environment**UNIT - III****Virtualization, Programming Models for Cloud Computing:** MapReduce, Cloud Haskell, Software Development in Cloud**UNIT - IV****Networking for Cloud Computing:** Introduction, Overview of Data Center Environment, Networking Issues in Data Centers, Transport Layer Issues in DCNs, Cloud Service Providers**UNIT - V**

Security in Cloud Computing, and Advanced Concepts in Cloud Computing

TEXT BOOK:

1. Chandrasekaran, K. *Essentials of cloud computing*. CRC Press, 2014

REFERENCE BOOKS:

1. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
2. Enterprise Cloud Computing - Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010
3. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010

CS 710 PE: AD-HOC & SENSOR NETWORKS (Professional Elective – IV)**B.Tech. IV Year I Sem.**

L	T	P	C
3	0	0	3

Prerequisites

1. Computer Networks
2. Distributed Systems
3. Mobile Computing

Course Objectives

- To understand the fundamental characteristics, real-world applications, and unique challenges associated with Mobile Ad Hoc Networks (MANETs).
- To evaluate the impact of broadcast and rebroadcast techniques in mitigating the broadcast storm problem and to compare the performance and efficiency of various multicasting protocols within MANET environments.
- To analyze geocasting mechanisms for both data delivery and route discovery, and to examine the limitations and enhancements of TCP protocols over dynamic MANET topologies.
- To gain comprehensive knowledge of Wireless Sensor Network (WSN) architecture, its classifications, and the critical physical layer issues that enable efficient sensor communications.
- To explore transport and application layer protocols in WSNs, focusing on adaptive, scalable, and energy-efficient techniques that address the inherent limitations and dynamic nature of sensor network environments.

Course Outcomes

- Understand the characteristics, applications, and challenges of Mobile Ad Hoc Networks (MANETs) and analyze the taxonomy and functionality of various topology-based and position-based routing algorithms.
- Evaluate broadcast and rebroadcast techniques to address the broadcast storm problem and compare various multicasting protocols in MANETs.
- Analyze geocasting mechanisms and their role in data and route-based transmissions and examine the issues and proposed solutions of implementing TCP over MANET environments.
- Identify the architecture, classifications, and physical layer concerns of wireless sensor networks and examine MAC, link, and routing layer protocols used in sensor communication.
- Explore the transport and application layer support mechanisms in wireless sensor networks and understand adaptive techniques suited for the dynamic nature of WSNs.

UNIT - I**Introduction to Ad Hoc Networks**

Characteristics of MANETs, Applications of MANETs and Challenges of MANETs.

Routing in MANETs

Criteria for classification, Taxonomy of MANET routing algorithms, *Topology-based* routing algorithms- Proactive: DSDV, WRP; Reactive: DSR, AODV, TORA; Hybrid: ZRP; *Position-based* routing algorithms- Location Services- DREAM, Quorum-based, GLS; Forwarding Strategies, Greedy Packet, Restricted Directional Flooding-DREAM, LAR; Other routing algorithms-QoS Routing, CEDAR.

UNIT - II**Data Transmission**

Broadcast Storm Problem, Rebroadcasting Schemes-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbour Knowledge-based: SBA, Multipoint Relaying, AHBP. Multicasting: Tree-based: AMRIS, MAODV; Mesh-based: ODMRP, CAMP; Hybrid: AMRoute, MCEDAR.

UNIT - III**Geocasting**

Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR.

TCP over Ad Hoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc

UNIT - IV

Basics of Wireless Sensors and Lower Layer Issues-Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

UNIT - V

Upper Layer Issues of WSN

Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

TEXT BOOKS

1. Ad Hoc and Sensor Networks – Theory and Applications, *Carlos Corderio Dharma P. Aggarwal*, World Scientific Publications, March 2006, ISBN – 981-256-681-3
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN – 978-1-55860-914-3 (Morgan Kauffman)

REFERENCE BOOKS:

1. C. Siva Ram Murthy, B.S. Manoj Ad Hoc Wireless Networks: Architectures and Protocols.
2. Taieb Znati Kazem Sohraby, Daniel Minoli, Wireless Sensor Networks: Technology, Protocols and Applications, Wiley.

CS 711 PE: ADVANCED ALGORITHMS (Professional Elective – V)**B.Tech. IV Year I Sem.**

L	T	P	C
3	0	0	3

Pre-Requisites: Algorithm Design and Analysis**Course Objectives:**

- To introduce advanced techniques in the design and analysis of algorithms, including amortized analysis and dynamic programming, to solve complex computational problems efficiently.
- To develop a strong understanding of graph algorithms, including graph matching, network flows, and shortest path algorithms, and their applications in real-world scenarios.
- To enable students to perform matrix computations and analyze flow networks, emphasizing algebraic approaches in algorithm design and analysis.
- To understand modular arithmetic and apply string-matching algorithms, including those based on hashing and finite automata, in algorithmic problem solving.
- To explore the concepts of randomized and approximation algorithms, and introduce fundamental topics in computational complexity.

Course Outcomes:

- Familiarize with advanced methods on analysis of algorithms
- Familiarize with the graphs, graph matching and shortest path algorithms
- Understand matrix computations and flow-networks
- Understand modulo representations and string algorithms
- Understand randomized, approximation algorithms and computational complexity topics

UNIT – I

Introduction to Algorithms, Classification of Algorithms, Asymptotic Analysis, Introduction to Recurrence equations - Linear recurrences, Non-linear recurrences, Formulation of recurrence equations, techniques for solving recurrence equations, Solving recurrence equations using polynomial reduction, Master's theorem

Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, Multistage Graph, topological sorting

UNIT – II

Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path, Bipartite matching problem

Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set, Optimal tree problems- optimal merge, Huffman coding, tree vertex splitting problem. **Shortest Path in**

Graphs: Floyd-Warshall algorithm, Travelling Sales Person Problem and introduction to dynamic programming paradigm. Optimal Graph Problems - Minimum Spanning Tree, Single source shortest path.

UNIT - III

Flow-Networks: Maxflow - mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

Matrix Computations: Strassen's algorithm and introduction to divide and conquer paradigm, Chain Matrix Multiplication, Matrix operations – Gaussian Elimination method, LUP-decomposition, Crout's method of decomposition, inverse of a triangular matrix,

UNIT - IV

Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation, interpolation problem. Multiplication of long integers by using Divide and Conquer paradigm, Schonhage-Strassen's Integer Multiplication algorithm.

String Algorithms: Naïve String, Rabin Karp, KMP, Boyer Moore, Harspool algorithms

UNIT - V

Basics of Computational Complexity: Introduction to computational complexity, complexity classes, Satisfiability problem and Cook's theorem, Examples of NP- Complete problems

Randomized algorithms: Introduction, Types of Randomized algorithms, Example of Randomized algorithms.

Approximation algorithms: Introduction, Types of Approximation algorithms, Examples of Approximation algorithms

TEXT BOOK:

1. Design and Analysis of Algorithms, S. Sridhar, Oxford University Press.

REFERENCE BOOKS:

1. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein.
2. The Design and Analysis of Computer Algorithms, Aho, Hopcroft, Ullman.
3. Algorithm Design, Kleinberg and Tardos.

CS 712 PE: AGILE METHODOLOGY (Professional Elective – V)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course Objectives:**

- Understand the foundational principles and practices of Agile methodology and Extreme Programming (XP) to enhance software development responsiveness and quality.
- Analyze the importance of real-time customer involvement and collaboration in Agile projects to ensure requirement clarity and continuous feedback.
- Demonstrate proficiency in managing software releases using version control systems and continuous deployment strategies to achieve efficient and reliable software delivery.
- Develop skills in risk identification, iteration planning, and sprint execution for effective Agile project management.
- Understand and apply concepts of incremental requirements gathering, code refactoring, and evolving design/architecture to support adaptive and maintainable software systems.

Course Outcomes:

- Identify basic concepts of agile methodology and Extreme programming
- Analyze real customer involvement in collaboration
- Demonstrate the ability to manage software releases through version for successful deployment cycles.
- Discuss risk management and iteration planning
- Understanding incremental requirements, refactoring, incremental design and architecture

UNIT - I**Introduction Extreme Programming (XP) - Agile Development**

Why Agile?, Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility, Agile methods-Scrum and XP, Manifesto for Agile Software Development, Principles of Agile Process. Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts, Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility, Practicing XP - Thinking, Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives.

UNIT - II**Collaborating**

Trust, Sit together, Real customer involvement, Ubiquitous language, Stand-Up meetings, coding standards, Iteration demo, Reporting.

UNIT - III**Releasing**

Bugfree Release, Version Control, Ten-Minute Build, continuous integration, Collective ownership and Documentation.

UNIT – IV**Planning**

Version, Release Planning, The Planning Game, Risk Management, Iteration Planning, Slack, Stories, and Estimating

UNIT - V**Developing**

Incremental requirements, Customer tests, Test driven development, Refactoring, Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing.

TEXT BOOK:

1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly, 2018.

REFERENCE BOOKS:

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004

CS 713 PE: ROBOTIC PROCESS AUTOMATION (Professional Elective – V)**B.Tech. IV Year I Sem.****L T P C****3 0 0 3****Course Objectives:**

- Understand the fundamental concepts, architecture, and benefits of Robotic Process Automation (RPA) and its role in modern business process automation.
- Apply flowchart and sequence-based logic to model and implement automation workflows for various business tasks and calculations.
- Gain hands-on experience with the UiPath platform, including the ability to debug, test, and troubleshoot automation processes effectively.
- Design and manage system-level automation techniques, including handling files, user input, and exception management to optimize workflow execution.
- Develop complete end-to-end applications for automating repetitive tasks using UiPath, integrating with external systems and databases as needed.

Course Outcomes:

- Understand the concepts of Robotic Process Automation.
- Apply the flow chart mechanism in various calculations.
- Applying UiPath tool for debugging process
- Design system managing techniques.
- Create application for process automation using UiPath tool.

UNIT - I

Robotic Process Automation: Introduction, Scope and techniques of automation, Robotic process automation, Components of RPA, RPA platforms, About UiPath

UiPath Stack UiPath Studio, UiPath Robot, Types of Robots, UiPath Orchestrator

UiPath Studio Projects, User interface

The User Interface: Task recorder, Advanced UI interactions: Input methods, Output methods

UNIT - II

Sequence, Flowchart, and Control Flow: Sequencing the workflow, Activities, Control Flow, various types of loops and decision making

Data Manipulation: Variables and scope, Collections, Arguments – Purpose and use, Data table usage with examples, File operation with step-by-step example, CSV/Excel to data table and vice versa

UNIT - III

Taking Control of the Controls: Finding and attaching windows, Finding the control, Techniques for waiting for a control, Act on controls – mouse and keyboard activities, Handling events, revisit recorder, When to use OCR, Types of OCR available, How to use OCR

Plugins and Extensions: Terminal Plugin, SAP Automation, Citrix automation and Credential management

UNIT - IV

Handling User Events and Assistant Bots: Assistant bots, Monitoring system event triggers, Monitoring image and element triggers, Launching an assistant bot on a keyboard event

Exception Handling, Debugging, and Logging: Exception handling, Common exceptions and ways to handle them, Logging and taking screenshots, Debugging techniques, Collecting crash dumps, Error reporting

UNIT - V

Managing and Maintaining the Code: Project organization, nesting workflows, Reusability of workflows, Commenting techniques, State Machine, When to use Flowcharts, State Machines, or Sequences, Using config files

Deploying and Maintaining the Bot: Publishing using publish utility, using Orchestration Server to control bots, deploy bots, License Management, Publishing and Managing updates

TEXT BOOK:

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath: Create Software robots. with the leading RPA tool – UiPath Kindle Edition

REFERENCE BOOK:

1. Robotic Process Automation A Complete Guide - 2020 Edition Kindle Edition.

CS 714 PE: BLOCKCHAIN TECHNOLOGY (Professional Elective – V)**B.Tech. IV Year I Sem.****L T P C****3 0 0 3****Prerequisites:**

- Understand the foundational concepts of cryptocurrency, including its role in blockchain ecosystems and digital finance.
- Explore the development and application of smart contracts for building decentralized applications (DApps).
- Examine public, private, and hybrid blockchain frameworks, their architectures, and use-case suitability.
- Analyze security, privacy, and regulatory challenges in blockchain systems, and understand performance and identity management in platforms like Bitcoin and Hyperledger Fabric.
- Design and implement blockchain-based solutions for real-world applications across domains such as finance, healthcare, education, IoT, and supply chain.

Course Objectives:

- To learn the fundamentals of Blockchain and various types of block chain and consensus mechanisms.
- To understand the public block chain system, Private block chain system and consortium blockchain.
- Able to know the security issues of blockchain technology.

Course Outcomes:

- Understanding concepts behind crypto currency
- Applications of smart contracts in decentralized application development
- Understand frameworks related to public, private and hybrid blockchain
- Analyze the security challenges and privacy aspects of blockchain technologies and evaluate their applications across domains such as finance, healthcare, education, and IoT.
- Create blockchain for different application case studies

UNIT-I

Fundamentals of Blockchain: Introduction, Origin of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain, The Technology and the Future.

Blockchain Types and Consensus Mechanism: Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol.

Cryptocurrency – Bitcoin, Altcoin and Token: Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrencies, Cryptocurrency Usage.

UNIT-II

Public Blockchain System: Introduction, Public Blockchain, Popular Public Blockchains, The Bitcoin Blockchain, Ethereum Blockchain.

Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry.

UNIT-III

Private Blockchain System: Introduction, Key Characteristics of Private Blockchain, Need of Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E-commerce Site Example, Various Commands (Instructions) in E-commerce Blockchain, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, Byzantine Fault, Multichain.

Consortium Blockchain: Introduction, Key Characteristics of Consortium Blockchain, Need of Consortium Blockchain, Hyperledger Platform, Overview of Ripple, Overview of Corda.

Initial Coin Offering: Introduction, Blockchain Fundraising Methods, Launching an ICO, Investing in an ICO, Pros and Cons of Initial Coin Offering, Successful Initial Coin Offerings, Evolution of ICO, ICO Platforms.

UNIT-IV

Security in Blockchain: Introduction, Security Aspects in Bitcoin, Security and Privacy Challenges of

Blockchain in General, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract (DApp), Security Aspects in Hyperledger Fabric.

Applications of Blockchain: Introduction, Blockchain in Banking and Finance, Blockchain in Education, Blockchain in Energy, Blockchain in Healthcare, Blockchain in Real-estate, Blockchain In Supply Chain, The Blockchain and IoT. Limitations and Challenges of Blockchain.

UNIT-V

Blockchain Case Studies: Case Study 1 – Retail, Case Study 2 – Banking and Financial Services, Case Study 3 – Healthcare, Case Study 4 – Energy and Utilities.

Blockchain Platform using Python: Introduction, Learn How to Use Python Online Editor, Basic Programming Using Python, Python Packages for Blockchain.

Blockchain platform using Hyperledger Fabric: Introduction, Components of Hyper ledger Fabric Network, Chain codes from Developer.ibm.com, Blockchain Application Using Fabric Java SDK.

TEXT BOOK:

1. “Blockchain Technology”, Chandramouli Subramanian, Asha A. George, Abhilasj K A and Meena Karthikeyan, Universities Press.

REFERENCE BOOKS:

1. Michael Juntao Yuan, Building Blockchain Apps, Pearson, India.
2. Blockchain Blueprint for Economy, Melanie Swan, SPD O'reilly.
3. Blockchain for Business, Jai Singh Arun, Jerry Cuomo, Nitin Gaur, Pearson.

CS 715 PE: SOFTWARE PROCESS & PROJECT MANAGEMENT (Professional Elective – V)**B.Tech. IV Year I Sem.****L T P C**
3 0 0 3**Course Objectives:**

- Introduce students to software process improvement frameworks such as CMMI, ISO/IEC standards, and other maturity models, emphasizing their importance in improving software quality and organizational capability.
- To provide knowledge of software economics and the software development life cycle
- Train students to identify and evaluate critical deliverables, performance metrics, key milestones (major and minor), and project artifacts essential for tracking and reporting project progress.
- Enable students to plan, manage, and execute software projects using conventional models (like Waterfall, Spiral) and modern approaches (like Agile, Scrum), balancing time, scope, and resources.
- To apply next-generation software project management practices and economics

Course Outcomes:

- Understand and evaluate the principles of software process maturity and various process reference models
- Gain knowledge of software economics, phases in the life cycle of software development, project organization, project control and process instrumentation
- Analyze the major and minor milestones, artifacts and metrics from management and technical perspective
- Design and develop software product using conventional and modern principles of software project management
- Understand and apply modern software project management practices and next-generation software economics to design and develop software products

UNIT - I**Software Process Maturity**

Software maturity Framework, Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The Optimizing Process, Process Reference Models Capability Maturity Model (CMM), CMMI, PCMM, PSP, TSP).

UNIT - II**Software Project Management Renaissance**

Conventional Software Management, Evolution of Software Economics, Improving Software Economics, Life-Cycle Phases and Process artifacts

Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model-based software architectures.

UNIT - III**Workflows and Checkpoints of process**

Software process workflows, Iteration workflows, Major milestones, minor milestones, periodic status assessments, Process Planning Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

UNIT - IV**Project Organizations**

Line-of- business organizations, project organizations, evolution of organizations, process automation. Project Control and process instrumentation, The seven-core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic software metrics, metrics automation.

UNIT - V

CCPDS-R Case Study and Future Software Project Management Practices, Modern Project Profiles, Next-Generation software Economics, Modern Process Transitions.

TEXT BOOKS:

1. Managing the Software Process, Watts S. Humphrey, Pearson Education
2. Software Project Management, Walker Royce, Pearson Education

REFERENCE BOOKS:

1. An Introduction to the Team Software Process, Watts S. Humphrey, Pearson Education, 2000
2. Process Improvement essentials, James R. Persse, O'Reilly, 2006
3. Software Project Management, Bob Hughes & Mike Cotterell, fourth edition, TMH, 2006
4. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006.
5. Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, 2nd edition, Wiley India, 2004.
6. Agile Project Management, Jim Highsmith, Pearson education, 2004.

CS 716 OE: OPERATING SYSTEMS (Open Elective –II)**B.Tech. IV Year I 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:

- To provide students with a comprehensive understanding of the structure, functionalities, and services of modern operating systems, including system calls, process, and thread management.
- To enable students to apply and evaluate CPU scheduling techniques and effectively use process management system calls, while understanding the causes, prevention, and recovery from deadlocks.
- To equip students with the ability to design and implement synchronization solutions and interprocess communication (IPC) mechanisms for maintaining process coordination and consistency.
- To help students analyze various memory management strategies, including paging, segmentation, and virtual memory, for improving system performance and resource utilization.
- To develop skills to work with file system structures and system calls, facilitating secure and efficient file handling within an operating system environment.
-

Course Outcomes:

- Understand the structure, services, and process management features of modern operating systems including system calls and thread concepts.
- Apply CPU scheduling algorithms and system calls for process management and analyze strategies to handle and recover from deadlocks.
- Implement synchronization techniques and interprocess communication mechanisms to ensure process coordination and data consistency.
- Analyze memory management techniques and virtual memory strategies to optimize system performance.
- Demonstrate file system operations and system calls for effective file management and protection in operating systems.

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.

REFERENCE BOOKS:

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

CS 717 OE: SOFTWARE ENGINEERING (Open Elective –II)**B.Tech. IV Year I Sem.**

L	T	P	C
3	0	0	3

Course Objectives

- To introduce students to the fundamental principles and practices of software engineering, including evolving software characteristics, layered technology, and software process models.
- To enable students to analyze, document, and manage software requirements effectively through the requirements engineering process, covering both functional and non-functional aspects.
- To develop the ability to design software architecture and models using established software design concepts and UML diagrams for representing both structural and behavioral views.
- To provide knowledge of software testing strategies, quality metrics, and debugging techniques essential for delivering reliable and validated software products.
- To equip students with the skills to identify, assess, and manage software risks, and to implement quality assurance practices and standards like ISO 9000 to ensure software excellence.

Course Outcomes

- Understand the fundamentals of software engineering, including software process models like Waterfall, Spiral, and Agile, and the role of process frameworks and CMMI.
- Analyze software requirements and apply the requirements engineering process to gather, specify, validate, and manage both functional and non-functional requirements.
- Design software architecture and models using standard design principles and notations such as UML diagrams to represent structural and behavioral aspects of software.
- Apply various software testing strategies and quality metrics to ensure software reliability and correctness through systematic validation and debugging processes.
- Evaluate and implement risk management and software quality assurance practices, including ISO standards, to enhance software quality and project success.

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI)

Process models: The waterfall model, Spiral model and Agile methodology

UNIT - II

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

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

UNIT - III

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

UNIT - IV

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

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

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 703 PC: CRYPTOGRAPHY AND NETWORK SECURITY LAB**B.Tech. IV Year I Sem.**

L	T	P	C
0	0	2	1

Course Objectives:

- Introduce students to software process improvement frameworks such as CMMI, ISO/IEC standards, and other maturity models, emphasizing their importance in improving software quality and organizational capability.
- To provide knowledge of software economics and the software development life cycle
- Train students to identify and evaluate critical deliverables, performance metrics, key milestones (major and minor), and project artifacts essential for tracking and reporting project progress.
- Enable students to plan, manage, and execute software projects using conventional models (like Waterfall, Spiral) and modern approaches (like Agile, Scrum), balancing time, scope, and resources.
- To apply next-generation software project management practices and economics

Course Outcomes:

- Apply the concepts of AND OR and XOR each character in this string and display the result.
- Design a Java program to perform encryption and decryption using various algorithms
- Implement and analyze modern symmetric key algorithms
- implement RC4 logic and encrypt text using the Blowfish algorithm
- Implement and evaluate public key cryptographic algorithms and message digest algorithms using Java and JavaScript

List of Experiments:

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and display the result.
2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.
3. Write a Java program to perform encryption and decryption using the following algorithms
 - a. Ceaser cipher b. Substitution cipher c. Hill Cipher
4. Write a C/JAVA program to implement the DES algorithm logic.
5. Write a C/JAVA program to implement the Blowfish algorithm logic.
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Write the RC4 logic in Java Using Java cryptography; encrypt the text "Hello world" using Blowfish. Create your own key using Java key tool.
8. Write a Java program to implement the RSA algorithm.
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
11. Calculate the message digest of a text using the MD5 algorithm in JAVA

TEXT BOOKS:

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 6th Edition
2. Cryptography and Network Security: Atul Kahate, McGraw Hill, 3rd Edition

REFERENCE BOOKS:

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

CS 704 PC: COMPILER DESIGN LAB**B.Tech. IV Year I Sem.**

L	T	P	C
0	0	2	1

Prerequisites

1. A Course on “Object Oriented Programming through Java”.

Co-requisites:

1. A course on “Web Technologies”.

Course Objectives:

- Train students to develop lexical analyzers that perform pattern matching, token recognition, and input buffering using regular expressions and LEX tool.
- Equip students with the skills to define grammar rules, resolve ambiguities, and build parsers using parser generators like YACC, including LL and LR parsing techniques.
- Provide hands-on experience in constructing abstract syntax trees (ASTs) and implementing semantic checks such as type compatibility and scope resolution.
- Teach students the mechanisms of symbol table creation and management, along with static, stack-based, and heap memory allocation strategies for compiler back-end components.
- To integrate lexical and syntax analyzers for building basic compilers or interpreters

Course Outcomes:

- Implement lexical analysis techniques including pattern matching and lexical token recognition using tools like LEX.
- Apply syntax analysis techniques using grammar rules, parser construction methods, and tools like YACC.
- Construct abstract syntax trees and perform semantic analysis including type checking.
- Analyze and implement storage allocation strategies and symbol table management.
- Integrate lexical and syntax analysis components to build mini compilers or interpreters using LEX and YACC.

List of Experiments

1. Implementation of symbol table.
2. Develop a lexical analyzer to recognize a few patterns inc (ex. Identifiers, constants, comments, operators etc.)
3. Implementation of lexical analyzer using lex tool.
4. Generate yacc specification for a few syntactic categories.
 - a) Program to recognize a valid arithmetic expression that uses operator +, -, * and /.
 - b) Program to recognize a valid variable which starts with a letter followed by any number of letter or digits.
 - c) Implementation of calculator using lex and yacc.
5. Convert the bnf rules into yacc form and write code to generate abstract syntax tree.
6. Implement type checking
7. Implement any one storage allocation strategies (heap, stack, static)
8. Write a lex program to count the number of words and number of lines in a given file or program.
9. Write a ‘C’ program to implement lexical analyzer using c program.
10. Write recursive descent parser for the grammar $E \rightarrow E+T \quad E \rightarrow T \quad T \rightarrow T^*F \quad T \rightarrow F \quad F \rightarrow (E)/id$.
11. Write recursive descent parser for the grammar $S \rightarrow (L) \quad S \rightarrow a \quad L \rightarrow L,S \quad L \rightarrow S$
12. Write a C program to calculate first function for the grammar $E \rightarrow E+T \quad E \rightarrow T \quad T \rightarrow T^*F \quad T \rightarrow F \quad F \rightarrow (E)/id$
13. Write a YACC program to implement a top down parser for the given grammar.
13. Write a YACC program to evaluate algebraic expression.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi

Sethi, Jeffry D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Loudon, Thomson.

MB 801 HS: ORGANIZATIONAL BEHAVIOUR**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives: This course demonstrates individual, group behavior aspects: The dynamics of organizational climate, structure and its impact on Organizations.

- Train students to develop lexical analyzers that perform pattern matching, token recognition, and input buffering using regular expressions and LEX tool.
- Equip students with the skills to define grammar rules, resolve ambiguities, and build parsers using parser generators like YACC, including LL and LR parsing techniques.
- Provide hands-on experience in constructing abstract syntax trees (ASTs) and implementing semantic checks such as type compatibility and scope resolution.
- Teach students the mechanisms of symbol table creation and management, along with static, stack-based, and heap memory allocation strategies for compiler back-end components.
- To integrate lexical and syntax analyzers for building basic compilers or interpreters

Course Outcomes:

- Understand the conceptual framework of the discipline of OB and its practical applications in the organizational set up.
- Apply different motivational theories and methods to increase the productivity and job satisfaction of employees.
- Understand communication in organizations and analyze participative decision-making techniques
- Understand the meaning and types of power, and evaluate the nature of groups
- Develop skills to design jobs, understand the principles of quality of work life and evaluate the process of behavioral modification within organizational contexts.

UNIT - I Organizational Behaviour

Definition, need and importance of organizational behaviour – Nature and scope – Frame work – Organizational behaviour models.

UNIT - II Individual Behaviour

Personality – types – Factors influencing personality – Theories – Learning – Types of learners – The learning process – Learning theories – Organizational behaviour modification, Misbehaviour – Types – Management Intervention. Emotions - Emotional Labour – Emotional Intelligence – Theories. Attitudes – Characteristics – Components – Formation – Measurement- Values. Perceptions – Importance – Factors influencing perception – Interpersonal perception- Impression Management. Motivation – importance – Types – Effects on work behavior.

UNIT - III Group Behaviour

Organization structure – Formation – Groups in organizations – Influence – Group dynamics – Emergence of informal leaders and working norms – Group decision making techniques – Team building - Interpersonal relations – Communication – Control.

UNIT - IV Leadership and Power

Meaning – Importance – Leadership styles – Theories of leadership – Leaders Vs Managers – Sources of power – Power centers – Power and Politics.

UNIT - V Dynamics of Organizational Behaviour

Organizational culture and climate – Factors affecting organizational climate – Importance. Job satisfaction – Determinants – Measurements – Influence on behavior. Organizational change – Importance – Stability Vs Change – Proactive Vs Reaction change – the change process – Resistance to change – Managing change. Stress – Work Stressors – Prevention and Management of stress – Balancing work and Life. Organizational development – Characteristics – objectives –. Organizational effectiveness

TEXT BOOKS:

1. Stephen P. Robins, Organisational Behavior, PHI Learning / Pearson Education, 11th edition, 2008.
2. Fred Luthans, Organisational Behavior, McGraw Hill, 11th Edition, 2001.

REFERENCE BOOKS:

1. Schermerhorn, Hunt and Osborn, Organisational behavior, John Wiley, 9th Edition, 2008.
2. Udai Pareek, Understanding Organisational Behaviour, 2nd Edition, Oxford Higher Education, 2004.

CS 803 PE: COMPUTATIONAL COMPLEXITY (Professional Elective – VI)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisite: Design and Analysis of Algorithms.**Course Objectives:**

- To develop a deep understanding of time and space complexity for various computational models and analyze the limits of algorithmic efficiency.
- To introduce the principles of optimization problems, helping students identify problem types and solution strategies within computational frameworks.
- To provide a solid foundation in NP-completeness theory, enabling students to classify problems and understand the implications of intractability.
- To explain hierarchy theorems and their role in illustrating the structured relationship between different complexity classes.
- To equip students with the knowledge to understand and analyze randomized computational models, including probabilistic Turing machines, and to evaluate complexity in randomized decision-making processes and graph-based problems.

Course Outcomes:

- Understand the complexity of time and space for computational models
- Understand optimizational problems
- Understand NP completeness problems
- Understand hierarchical theorems
- Understand and analyze randomized computational models and evaluate the complexity of decision-making using randomized decision trees and graph-based properties.

UNIT – I

Introduction: Algorithms and complexity, Basic Complexity Classes-Deterministic time and the class P. Computational Tasks and models: Computational tasks – Search problems, Decision problems, Uniform models- Overview, General Principles, Concrete Model, Halting problem, restricted models.

UNIT – II

P vs. NP: Efficient Computation, The Search Version (Finding vs. Checking), The Decision Version (Proving Vs Verifying), Equivalence of the two formulations, Optimal Search Algorithms for NP Polynomial time reduction: The general notation of a Reduction, Reducing Optimization Problems to search problems, Self-Reducibility of search problems

UNIT – III

NP – Completeness: Definition, Cook's theorem, Existence of NP Complete Problems bounded halting and non-halting, Natural NP Complete Problems – The NP completeness of CSAT, The NP Completeness of SAT, Combinatorics and Graph Theory, additional properties of the standard reductions, Negative applications of NP Completeness, Positive applications of NP Completeness, NP Sets, Reflections on Complete problems, NP – complete optimization problems.

UNIT –IV

Diagonalization: Time Hierarchy theorem, Space Hierarchy theorem, Non-deterministic Time Hierarchy theorem, Ladner's theorem.

Space Complexity: Definition of space bounded computation, PSPACE completeness, NL Completeness, some space complexity classes– Savitch's theorem, Savitch's theorem, The essence of PSPACE

The polynomial time hierarchy and alternations: polynomial hierarchy, time versus alternations, properties of polynomial hierarchy, Complete problems in PH.

UNIT – V

Randomized computation: Probabilistic Turing machine, one sided and zero-sided error, Randomized

reduction, Randomized space bounded computation.

Decision trees: Graphs and Decision Trees, Monotonic Graph properties, Topological criterion, Randomized decision trees.

TEXT BOOKS:

1. The Basics of Computational Complexity, Oded Goldreich, Cambridge University Press
2. Computational Complexity: A Modern Approach, Sanjeev Arora and Boaz Barak, Princeton University

REFERENCE BOOKS:

1. Computational Complexity, by Christos Papadimitriou
2. Theory of Computational Complexity, Ding-Zhu Du, Ker-I Ko, WILEY

CS 804 PE: DISTRIBUTED SYSTEMS (Professional Elective –VI)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites:

1. A course on “Operating Systems”.
2. A course on “Computer Organization & Architecture”.

Course Objectives:

- To introduce the fundamental principles, challenges, and architecture of distributed systems, emphasizing interprocess communication models and remote invocation techniques.
- To enable students to analyze operating system support, including process and communication models, and understand the architecture of distributed file systems.
- To develop the ability to evaluate peer-to-peer system architectures and implement coordination techniques like clock synchronization, distributed mutual exclusion, and consensus protocols.
- To equip students with the knowledge to apply transaction models and concurrency control techniques for maintaining data integrity in distributed environments, along with atomic commit protocols and deadlock recovery.
- To help students understand and design replication strategies and distributed shared memory systems, focusing on consistency models and fault-tolerant communication mechanisms.

Course Outcomes:

- Understand the fundamental characteristics and models of distributed systems, including interprocess communication and remote invocation techniques such as Java RMI.
- Analyze the role of operating system support and file service architectures in the design and functioning of distributed systems.
- Evaluate peer-to-peer architectures and implement coordination mechanisms like clock synchronization, distributed mutual exclusion, and consensus protocols.
- Apply transaction models and concurrency control mechanisms in distributed environments and analyze protocols for atomic commit and distributed deadlock handling.
- Understand the concepts of replication and distributed shared memory, including consistency models and fault-tolerant communication strategies.

UNIT - I

Characterization of Distributed Systems: Examples of Distributed systems, Resource sharing and web, challenges

System models: Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication

Distributed objects and Remote Invocation: Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT - II

Operating System Support- OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture.

Distributed File Systems-Introduction, File Service architecture.

UNIT - III

Peer to Peer Systems– Napster and its legacy, Peer to Peer middleware

Time and Global States-Introduction, Clocks, events and Process states, Synchronizing physical clocks, logical time and logical clocks, global states, distributed debugging.

Coordination and Agreement- Distributed mutual exclusion, Elections, Multicast communication, consensus and related problems.

UNIT - IV

Transactions and Concurrency Control- Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering.

Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions

Distributed deadlocks: Transaction recovery.

UNIT - V

Replication: Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data.

Distributed shared memory: Design and Implementation issues, Consistency models.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

CS 805 PE: DEEP LEARNING (Professional Elective –VI)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives:

- To provide foundational knowledge of machine learning principles and the architecture and functioning of neural networks.
- To enable students to understand and apply techniques for the optimal preparation and utilization of data when training deep learning models.
- To develop the ability to apply Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for solving real-world data-driven problems.
- To equip students with the skills to evaluate deep learning models using appropriate performance metrics and validation techniques.
- To guide students in the design and development of deep learning models tailored to address real-world challenges across various domains.

Course Outcomes:

- Understand machine learning basics and neural networks
- Understand optimal usage of data for training deep models
- Apply CNN and RNN models for real-world data
- Evaluate deep models
- Develop deep models for real-world problems

UNIT -I**Machine Learning Basics**

Learning Algorithms, Capacity, Overfitting and Underfitting, Hyperparameters and Validation Sets, Estimators, Bias and Variance, Maximum Likelihood Estimation, Bayesian Statistics, Supervised Learning Algorithms, Unsupervised Learning Algorithms, Stochastic Gradient Descent, Building a Machine Learning Algorithm, Challenges Motivating Deep Learning

Deep Feedforward Networks Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms

UNIT -II**Regularization for Deep Learning**

Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised Learning, Multi- Task Learning, Early Stopping, Parameter Tying and Parameter Sharing, Sparse Representations, Bagging and Other Ensemble Methods, Dropout, Adversarial Training, Tangent Distance, Tangent Prop, and Manifold Tangent Classifier, Optimization for Training Deep Models, Learning vs Pure Optimization, Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates

UNIT-III**Convolutional Networks**

The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Structured Outputs, Data Types, Efficient Convolution Algorithms, Random or Unsupervised Features

UNIT -IV**Recurrent and Recursive Nets**

Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Echo State Networks, Leaky Units and Other Strategies for Multiple Time Scales, The Long Short-Term Memory and Other Gated RNNs, Optimization for Long- Term Dependencies, Explicit

Memory

UNIT -V

Practical Methodology: Performance Metrics, Default Baseline Models, Determining Whether to Gather More Data, Selecting Hyperparameters, Debugging Strategies, Example: Multi-Digit Number Recognition

Applications: Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing, Other Applications.

TEXT BOOK:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.

REFERENCE BOOKS:

1. The Elements of Statistical Learning. Hastie, R. Tibshirani, and J. Friedman, Springer.
2. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.
3. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006.
4. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
5. Golub, G., H., and Van Loan, C.,F., Matrix Computations, JHU Press, 2013.
6. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

CS 806 PE: HUMAN COMPUTER INTERACTION (Professional Elective –VI)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Course Objectives:

- To gain an overview of Human-Computer Interaction (HCI)
- Understanding the alternatives to traditional "keyboard and mouse" computing.
- To develop an understanding of the social, legal, and ethical implications of technology, and to instill a sense of professional and ethical responsibility in engineering practice.
- To emphasize the significance of design and evaluation methodologies in the development of responsible, user-centered, and effective engineering solutions.
- Be able to apply models from cognitive psychology to predicting user performance

Course Outcomes:

- Apply HCI and principles to interaction design.
- Design certain tools for blind or PH people
- Understand the social implications of technology and ethical responsibilities as engineers.
- Understand the importance of a design and evaluation methodology
- Understand cognitive model goals and design focus

UNIT - I

Introduction: Importance of user Interface – definition, importance of good design. Benefits of good design, A brief history of Screen design.

The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

UNIT - II

Design process – Human interaction with computers, importance of human characteristics, human consideration, Human interaction speeds, understanding business junctions.

Screen Designing: Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT- III

Windows – New and Navigation schemes selection of window, selection of devices based and screen- based controls. Components – text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

UNIT- IV

HCI in the software process- The software life cycle, Usability engineering, Iterative design and prototyping, Design Focus: Prototyping in practice, Design rationale, Design rules, Principles to support usability Standards, Golden rules and heuristics, HCI patterns, Evaluation techniques, Goals of evaluation, Evaluation through expert analysis, Evaluation through user participation, Choosing an evaluation method, Universal design, Universal design principles Multimodal interaction

UNIT- V

Cognitive models Goal and task hierarchies Design Focus: GOMS saves money, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures, Ubiquitous computing and augmented realities, Ubiquitous computing applications research, Design Focus: Ambient Wood – augmenting the physical, Virtual and augmented reality, Design Focus: Shared experience Design Focus: Applications of augmented reality Information and data visualization

TEXT BOOKS:

1. The essential guide to user interface design, Wilbert O Galitz, Wiley Dream Tech.
2. Human – Computer Interaction. Alan Dix, Janet Fincay, Gregory's, Abowd, Russell Bealg, Pearson Education.

REFERENCE BOOKS:

1. Designing the user interface. 3rd Edition Ben Shneidermann, Pearson Education Asia.
2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
3. User Interface Design, Soren Lauesen, Pearson Education.
4. Human –Computer Interaction, D. R. Olsen, Cengage Learning.
5. Human –Computer Interaction, Smith - Atakan, Cengage Learning.

CS 807 PE: CYBER FORENSICS (Professional Elective –VI)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites: Network Security.**Course Objectives:**

- Provide students with foundational knowledge about various forms of cybercrime, principles of digital forensics, and procedures involved in effective incident response during cyber investigations.
- Enable students to perform appropriate initial response activities, including volatile data collection and imaging techniques, to preserve digital evidence in a forensically sound manner.
- Train students in using analytical methods and tools for digital evidence analysis, including validation techniques and methodologies specific to network forensics and intrusion detection.
- Provide hands-on experience with leading forensic tools (e.g., FTK, Autopsy, EnCase, Cellebrite) to perform investigations on computers, mobile devices, and email systems effectively.
- Learn file system structures and whole disk encryption metho

Course Outcomes:

- Understand the fundamental knowledge of cybercrime, digital forensics principles, and effective incident response procedures in computer-related criminal investigations.
- Understand the initial response techniques, volatile data collection, and the critical process of forensic duplication in digital forensic investigations.
- Ability to gain comprehensive knowledge and practical skills in digital forensics analysis, validation techniques, and network forensics methodologies
- Ability to evaluate, utilize, and apply current forensic tools in computer mobile device, investigations and email-related investigations
- Understand file systems and develop knowledge of whole disk encryption techniques

UNIT- I

Introduction of Cybercrime: Types, The Internet spawns crime, Worms versus viruses, Computers' roles in crimes, Introduction to digital forensics, Introduction to Incident - Incident Response Methodology – Steps - Activities in Initial Response, Phase after detection of an incident

UNIT- II

Initial Response and forensic duplication, Initial Response & Volatile Data Collection from Windows system - Initial Response & Volatile Data Collection from Unix system – Forensic Duplication: Forensic duplication: Forensic Duplicates as Admissible Evidence, Forensic Duplication Tool Requirements, Creating a Forensic Duplicate/Qualified Forensic Duplicate of a Hard Drive

UNIT- III

Forensics analysis and validation: Determining what data to collect and analyze, validating forensic data, addressing data-hiding techniques, performing remote acquisitions
 Network Forensics: Network forensics overview, performing live acquisitions, developing standard procedures for network forensics, using network tools, examining the honeynet project.

UNIT- IV

Current Forensic tools: evaluating computer forensic tool needs, computer forensics software tools, computer forensics hardware tools, validating and testing forensics software
 E-Mail Investigations: Exploring the role of e-mail in investigation, exploring the roles of the client and server in e-mail, investigating e-mail crimes and violations, understanding e-mail servers, using specialized e-mail forensic tools.

Cell phone and mobile device forensics: Understanding mobile device forensics, understanding acquisition procedures for cell phones and mobile devices.

UNIT- V

Working with Windows and DOS Systems: understanding file systems, exploring Microsoft File Structures,

Examining NTFS disks, Understanding whole disk encryption, windows registry, Microsoft startup tasks, MS-DOS startup tasks, virtual machines.

TEXT BOOKS:

1. Kevin Mandia, Chris Prosise, "Incident Response and computer forensics", Tata McGraw Hill, 2006.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi.
3. Computer Forensics and Investigations by Nelson, Phillips Enfinger, Steuart, CENGAGE Learning

REFERENCE BOOKS:

1. Real Digital Forensics by Keith J. Jones, Richard Bejtlich, Curtis W. Rose, Addison- Wesley Pearson Education
2. Forensic Compiling, A Tractitioneris Guide by Tony Sammes and Brian Jenkinson, Springer International edition.

CS 808 OE: ALGORITHMS DESIGN AND ANALYSIS (Open Elective –III)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites: Programming for problem solving and Data Structures**Course Objectives:**

- Introduces the notations for analysis of the performance of algorithms.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
- Describes how to evaluate and compare different algorithms using worst, average, and best case analysis.
- Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

Course Outcomes:

- Analyze the performance of algorithms
- Apply disjoint set operations and heap-based priority queues, and implement backtracking techniques to solve combinatorial problems such as N-Queens, graph coloring, and Hamiltonian cycles.
- Use dynamic programming strategies to solve optimization problems like the 0/1 knapsack, all-pairs shortest path, and traveling salesman problem efficiently.
- Apply greedy algorithms to real-world problems including job sequencing, minimum spanning trees, and shortest paths, and utilize traversal techniques for analyzing trees and graphs.
- Implement branch and bound techniques for solving complex problems and understand the concepts of NP-Hard and NP-Complete problems with reference to non-deterministic algorithms and Cook's theorem.

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-queen's 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 sales person 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 sales person 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 Rajasekharan, University Press.

REFERENCE BOOKS:

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.

CS 809 OE: INTRODUCTION TO COMPUTER NETWORKS (Open Elective – III)**B.Tech. IV Year II Sem.**

L	T	P	C
3	0	0	3

Prerequisites

1. A course on “Programming for problem solving”
2. A course on “Data Structures”

Course Objectives

- Equip the students with the concepts and fundamentals of computer networks.
- To provide foundational knowledge of network architectures, focusing on the functions and responsibilities of each layer in the OSI and TCP/IP reference models.
- To help students understand and apply concepts of IP addressing, subnetting, and routing mechanisms used in network communication.
- To explore the services and operations of the transport layer, with a focus on TCP and UDP protocols, including aspects of connection management, flow control, and reliability.
- To introduce and explain the protocols and services at the application layer, such as DNS, email, HTTP, and multimedia streaming, and to evaluate their performance in real-world networking scenarios.

Course Outcomes

- Gain the knowledge of the basic computer network technology.
- Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
- Understand subnetting and routing mechanisms.
- Understand the services and functionalities of the transport layer and analyze the working of TCP and UDP protocols, including connection management and reliability mechanisms.
- Explain the role of the application layer protocols and services such as DNS, email, HTTP, and multimedia streaming, and evaluate their operation in real-world network communication.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless Transmission.

Data link layer: Design issues, framing, Error detection and correction.

UNIT - II

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V

Application Layer: Domain name system, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 6th Edition. Pearson Education

REFERENCE BOOKS:

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