SHIV NADAR
UNIVERSITY
CHENNAI

SSN School of Engineering

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

B.TECH.
COMPUTER SCIENCE AND ENGINEERING

CURRICULUM & SYLLABUS

REGULATIONS 2025

PO PSO

PO 1: Engineering knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO 2: Problem analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO 3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9**: Communication**: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)
PSO 1: Design and develop reliable computing solutions by applying algorithmic thinking, programming paradigms and Software Engineering practices.
PSO 2: Construct efficient, scalable, and secure real-world computing systems by integrating software, hardware, and embedded architectures, leveraging emerging technologies

COURSE SUMMARY

The listed courses in the curriculum are broadly classified as per the recommendations from the UGC.

Sl. No	Broad Category of Course	Minimum Credit Requirement
1	Major Core (MC)	80
2	Minor Stream (MS)	32
3	Multidisciplinary (MD)	11
4	Ability Enhancement Course (AEC)	08
5	Skill Enhancement Course (SEC)	07
6	Value Added Courses (VAC)	08
7	Summer Internship (SI)	02
8	Project Dissertation (PD)	12
	Total	160

The semester wise credit breakup of the Curriculum based on the above credit breakup proposed by the UGC is as follows: **(Revised)**

Semester\Category	**MC**	**MS**	**MD**	**AEC**	**SEC**	**VAC**	**SI**	**PD**	**EAA#**	**TOTAL**
1	9		9	3					1	**21 (+1)**
2	14			3		2			1	**20 (+1)**
3	15	7			1					**23**
4	14	3			5					**22**
5	7	9		2	2	2				**22**
6	11	6						4		**21**
7	10	5					2	6		**23**
8		2				4		2		**8**
Total	**80**	**32**	**9**	**8**	**9**	**8**	**2**	**12**	**2**	**160 (+2)**

#Extra Academic activities (EAA) are not included in the CGPA calculation.

SEMESTER I

Sl. No.	Course Title	Course Category	Contact Periods	L	T	P	C
1	Communicative English	AEC	3	2	1	0	3
2	Mathematics I	MD	4	3	1	0	4
3	Engineering Physics	MD	3	3	0	0	3
4	Engineering Physics LAB	MD	4	0	0	4	2
5	Problem Solving and Programming in C	MC	6	2	0	4	4
6	Basic Electrical Technology	MC	2	2	0	0	2
7	Digital Design + LAB	MC	4	2	0	2	3
8	Extra Academic Activity 1#	EAA	2	0	0	0	1
Total			**29**	**13**	**2**	**12**	**21**

SEMESTER II

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Technical Writing	AEC	3	2	1	0	3
2	Mathematics II	MC	3	3	0	0	3
3	Data Structures	MC	3	3	0	0	3
4	Computer Organization and Architecture	MC	3	3	0	0	3
5	Object Oriented Programming + LAB	MC	4	2	0	2	3
6	Environmental Science and Engineering	VAC	2	2	0	0	2
7	Data Structures Lab	MC	4	0	0	4	2
8	Unix Scripting Environment	SEC	2	0	0	2	1
9	Extra Academic Activity 2#	EAA	2	0	0	0	1
Total			**26**	**17**	**1**	**6**	**20**

SEMESTER III

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Design and Analysis of Algorithms + LAB	MC	7	3	0	4	5
2	Database Management Systems	MC	3	3	0	0	3
3	Linear Algebra	MS	3	3	0	0	3
4	Java Programming + LAB	MS	6	2	0	4	4
5	Operating Systems +LAB	MC	7	3	0	4	5
6	Interactive Engineering graphics	SEC	2	0	0	2	1
7	Database Management Systems Lab	MC	4	0	0	4	2
Total			**32**	**14**	**0**	**18**	**23**

SEMESTER IV

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Computer Networks + LAB	MC	7	3	0	4	5
2	Artificial Intelligence + LAB	MC	4	2	0	2	3
3	Discrete Mathematics	MC	3	3	0	0	3
4	Full Stack Web Technologies	MC	3	3	0	0	3
5	Full Stack Web Technologies LAB	SEC	4	0	0	4	2
6	Embedded Programming + LAB	MS	4	2	0	2	3
7	Engineering Practices Laboratory	SEC	2	0	0	2	1
8	Small Scale Application Development (MINI PROJECT)	SEC	3	0	1	2	2
Total			30	13	1	16	22

SEMESTER V

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Quantum Computing Fundamentals	MS	3	3	0	0	3
2	Machine Learning + LAB	MC	5	3	0	2	4
3	Theory of Computation	MC	3	3	0	0	3
4	MINOR STREAM COURSE 1	MS	3	3	0	0	3
5	MINOR STREAM COURSE 2	MS	3	3	0	0	3
6	Ability Enhancement Course Basket	AEC	2	2	0	0	2
7	AI-Assisted Software Development and Collaboration	SEC	4	0	0	4	2
8	Value Added Course Basket	VAC	2	2	0	0	2
Total			**25**	**19**	**0**	**6**	**22**

SEMESTER VI

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Computer Security + LAB	MC	5	3	0	2	4
2	Computer Graphics + LAB	MC	5	3	0	2	4
3	Software Engineering + LAB	MC	4	2	0	2	3
4	MINOR STREAM COURSE 3	MS	3	3	0	0	3
5	MINOR STREAM BASKET 1	MS	3	3	0	0	3
6	Capstone Project (Phase 1 – Ideation and Literature Review)	PD	8	0	0	8	4
Total			**28**	**14**	**0**	**14**	**21**

SEMESTER VII

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Big Data Systems + LAB	MC	5	3	0	2	4
2	Distributed Systems	MC	3	3	0	0	3
3	Compiler Design + LAB	MC	4	2	0	2	3
4	MINOR STREAM BASKET 2	MS	3	3	0	0	3
5	MINOR STREAM BASKET 3	MS	2	2	0	0	2
6	Capstone Project (Phase 2 – Prototype)	PD	12	0	0	12	6
7	Industrial Training/Internship	SI	4	0	0	4	2
Total			**33**	**13**	**0**	**20**	**23**

SEMESTER VIII

Sl. No	Course Title	Course Category	Contact Periods	L	T	P	C
1	Value Added Course Basket	VAC	2	2	0	0	2
2	Professional Ethics and Human Values	VAC	2	2	0	0	2
3	MINOR STREAM BASKET 4	MS	2	2	0	0	2
4	Capstone Project (Phase 3)	PD	4	0	0	4	2
Total			**10**	**6**	**0**	**4**	**8**

MINOR STREAMS

The Department of Computer Science and Engineering offers the following list of Minor Streams.

1. **Quantum Computing**
2. **High performance Computing**
3. **Computational Intelligence**

Each Minor Stream has seven courses worth 19 credits. The student must enroll in any one of the three Minor Streams.

Minor Stream 1- Quantum Computing

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Mathematical Foundations of Quantum Computing	5	3	3	0	0	3
2	Quantum Algorithms & Complexity + LAB	5	3	2	0	2	3
3	Quantum Machine Learning	6	3	3	0	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	2	2	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	3	3	0	0	2

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (2 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (3 Credit courses)***
Quantum Programming and Simulation	Quantum Information Theory	Quantum Sensor Networks and IoT	Quantum Cryptography
Principles of Quantum Mechanics for Computing	Quantum Error Correction and Fault Tolerance	Hybrid Quantum–Classical Computing	Quantum Networking
Quantum Optimization	Quantum Computer Architecture	Quantum Circuit Simulation and Tensor Networks	Advanced Quantum Machine Learning

Minor Stream 2- Advanced Computing Technologies

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Heterogeneous Computing system	5	3	3	0	0	3
2	Spatial Computing	5	3	3	0	0	3
3	Cloud Computing	6	3	3	0	0	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	2	2	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	3	3	0	0	2

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Mobile and Pervasive Computing	Edge Computing and IoT	Software Defined Networking	Sensors and Actuators for Human-Computer Interaction
Parallel Architectures and Programming	Real time OS	GPU Computing	Affective Computing
Neuromorphic Computing	Immersive Technologies	Randomized Algorithms	Functional Programming
AI-assisted Software Engineering	Agile Methodologies and DevOps	UI/UX Design	Software Architecture and Design Patterns
Digital Forensics			

Minor Stream 3- Computational Intelligence

Sl. No.	Course Title	Semester	Contact Hours	L	T	P	C
1	Deep learning concepts and applications	5	3	3	0	0	3
2	Image and video processing	5	3	2	0	2	3
3	Natural Language processing	6	3	2	0	2	3
4	MINOR STREAM BASKET 1	6	3	3	0	0	3
5	MINOR STREAM BASKET 2	7	2	2	0	0	3
6	MINOR STREAM BASKET 3	7	2	2	0	0	2
7	MINOR STREAM BASKET 4	8	3	3	0	0	2

The following is the tentative list of courses that may be offered under the ***Minor Stream Baskets***:

Minor Stream Basket 1 (3 Credit courses)	***Minor Stream Basket 2 (3 Credit courses)***	***Minor Stream Basket 3 (2 Credit courses)***	***Minor Stream Basket 4 (2 Credit courses)***
Modern Computer Vision	Large Language Models and Generative AI	AI Ethics and Fairness	Medical Image Processing
Applied Optimization Techniques	Edge AI for Low Power Devices	Applied Agentic AI	Robotics and Autonomous Systems
Reinforcement Learning	Knowledge Engineering	Social Network Analytics	Health Care Analytics
Multi-modal Machine Learning	Time Series Analysis and Forecasting	Business Intelligence and Analytics	Advanced Speech Technology

MICROSPECIALIZATION

The University proposes to offer Micro-Specializations to Undergraduate students. Its salient features are as follows:

1. Each Micro-Specialization has a defined structure in terms of three sequential components:
    - **Component I** - One Foundation course that constitutes a mandatory requirement and also a prerequisite for subsequent components.
    - **Component II** - Two courses from a specified vertical.
    - **Component III** - Project/Design/Term Paper
2. A Student would be required to complete all three components (10-12 credits) from the specified stream to earn a Micro-Specialization.

Eligibility for Micro-Specialization registration

To register for a Micro-Specialization, the student must have completed all curricular requirements up to the previous semester and have a CGPA $\geq$ 8.5. The student must maintain a CGPA $\geq$ 8.5 without any backlog in the subsequent semesters to keep the Micro-Specialization registration active. Micro-Specialization must not overlap with the Major Core courses or Minor Stream courses credited as part of the degree program.

Provisional List of Micro-Specialization Verticals

The following is the list of Micro-Specialization verticals offered to our undergraduate students:

Micro-Specialization Verticals	**Component I**	**Component II**		**Component III**
Visual Computing	Introduction to Visual Computing	Computer Vision and Image Understanding	3D Vision and Graphics Systems	Project / Design / Term Paper
Reliable Autonomous Systems & Perception	Autonomous Systems	Aerial Robotics and Perception	Autonomous Mobile Robots	
Cyber Security and Ethical Hacking.	System Security	Application Security	Ethical Hacking and Network Security	
Language and Speech Technologies	Foundations of Language Technologies	Neural Architectures for Text and Speech Processing	Modern Language Technologies and Applications	

Value Added Course Basket

Sl. No.	Course Title	C
1	Green Computing	2
2	Financial Data Analysis	2
3	Business Intelligence	2
4	Water Resource Management	2
5	Entrepreneurship Essentials	2
6	Version Control and Collaboration Lab	2

Ability Enhancement Course Basket

Sl. No.	Course Title	C
1	Academic Writing	3
2	Linguistics	3
3	Mass Communication	2

SEMESTER I

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	AEC	COMMUNICATIVE ENGLISH	2	1	0	3	3

OBJECTIVES:

The learners will be able to:

- Enhance conversation fluency and networking skills in English.
- Contribute efficiently to meetings and group discussions.
- Deliver well-structured presentations and participate in interviews.
- Learn the language and strategies of successful negotiations.
- Communicate effectively during conflict-resolving situations.

UNIT I CONVERSATION AND NETWORKING SKILLS 6 + 3

Introducing yourself, peers and others. Conversation skills: starting a conversation; keeping the conversation going; closing the conversation; giving directions and instructions; learning functional language needed for everyday interactions. Imparting the basic language skills and techniques for telephonic conversations. Role-plays as a technique for practicing conversation. Writing professional emails. Reading short texts on everyday professional interactions.

UNIT II MEETING AND GROUP DISCUSSION SKILLS 6 + 3

Setting up, organizing and conducting effective meetings; learning the language used to lead a meeting; practicing how to agree and disagree with ideas and people politely. Reading sample minutes of meetings and pre-meeting preparation, and mock meetings. Making notes, summarizing, and reporting in meetings. Group discussion skills: Basic skills required for a GD; strategies for improving discussion skills; learning the language used for a GD; voicing an opinion and arguing a point effectively, leading a discussion; chairing a discussion; discussing to arrive at a consensus. Pre-GD preparation, reading GD transcripts and mock GDs for practice.

UNIT III PRESENTATION AND INTERVIEW SKILLS 6 + 3

Making effective presentations: structure of a presentation; referring to visual data; describing visuals; non-verbal communication skills; examples of model presentations; and learning the language used to deliver effective and persuasive presentations. Advance topic research, mock presentations and reading sample presentations for practice. Applying for jobs; writing resumes; and cover letters. Participating in job interviews and other interviews. Strategies for effective interviewing: preparing for the interview, during-the-interview and after-the-interview etiquette. Common interview questions and learning interview language skills. Mock interviews and reading interview transcripts for practice.

UNIT IV NEGOTIATION SKILLS AND ASSERTIVENESS 6 + 3

Negotiation strategies; negotiating effectively; exchanging and comparing information; building rapport with people; practicing the language used in negotiations; considering and respecting cultural differences while negotiating; making compromises. Communication styles: passive, aggressive and assertive. Reading case studies on cross-cultural negotiation scenarios and mock negotiation roleplays.

UNIT V CONFLICT RESOLUTION AND DECISION-MAKING SKILLS 6 + 3

Resolving conflicts: listening actively to ideas; handling people smoothly; dealing with and attempting to solve conflicting problems. Making a decision: thinking about various possibilities before decision-making; communicating the pros and cons of a decision; presenting a mutually beneficial solution. Reading case studies and mock conflict resolution, decision making role plays. Developing cross-cultural communication skills.

Lecture	**: 30 Periods**
Tutorial	**: 15 Periods**
Total	**: 45 Periods**

COURSE OUTCOMES:

On successful completion of this course, the students will be able to:

	Course Outcomes
CO1	Develop conversation skills and networking skills in English
CO2	Communicate effectively in meetings and group discussions
CO3	Deliver engaging presentations and professional interviews
CO4	Speak and negotiate strategically in formal and informal settings
CO5	Present effectively during conflict resolving situations

TEXTBOOKS:

1. Bovee, C. L.& Thill, J. V. (2016). *Business Communication Today, 15/e*. Pearson Education India.
2. Beebe, S. A., & Mottet, T. P. (2021). *Business and Professional Communication (3rd ed)*. Pearson Higher Education AU.
3. Herbert Hirsch. (2007). Essential Communication Strategies: For Scientists, Engineers, and Technology Professionals, 2nd edition. Wiley-IEEE Press.

REFERENCES:

1. Dodd, C. H. (2012). Managing business and professional communication (3rd ed). Allyn & Bacon/Pearson.
2. Dwyer, J. (2012). *Communication for Business and the Professions: Strategies and Skills* (7th ed). Pearson Higher Education AU.
3. Gill Hasson. (2012). Brilliant Communication Skills: What the Best Communicators Know, Do and Say. Ft Pr.
4. Jones, L., & Alexander, R. (2011). *New international business English updated edition student's book: Communication skills in English for business purposes* (Vol. 3). Cambridge university press.
5. Kerry Patterson, Joseph Grenny, Ron McMillan and Al Switzler. (2013). Crucial Conversations: Tools for Talking When Stakes Are High. Brilliance Audio.

Course code	Category	Mathematics I	L	T	P	C	Contact Hours
MA1039	MD		3	1	0	4	4

Course Outcomes:

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

CO1: Understand and apply fundamental concepts of differential and integral calculus.

CO2: Apply the concepts of integration to analyze the vector fields.

CO3: Apply various test methods to study the convergence of sequences and series.

CO4: Solve various types of first-order ordinary differential equations.

CO5: Understand functions of two variables and apply them to solve optimization problems.

Course Objectives:

This course aims to:

1. Introduce the concepts of limits, continuity, and differentiation, including an introduction to partial derivatives and gradients.
2. Cover the techniques of integration and their applications.
3. Familiarize with the sequences, series, and power series.
4. Introduce the basic concepts, classifications, and methods for solving first-order ordinary differential equations.
5. Cover methods for solving higher-order ordinary differential equations and introduce optimization with constraints using Lagrange multipliers.

Syllabus:

Unit 1 Calculus **12**

Limits, Continuity, Increment Theorem, Chain Rules, Extreme Values, Antiderivatives - The Definite Integral - The Fundamental Theorem of Calculus - Techniques of Integration (Substitution, Partial Fractions, Integration by Parts, Bernoulli's formula), Applications of Integration (Area, Volume, Arc Length), Surface Area of solids of Revolution - Improper Integrals.

Unit 2 Vector Calculus **12**

Vectors in 2-space and 3-space, Scalar product, Vector product, Scalar and vector functions, Curl and Divergence of a vector field, Level curves and surfaces, Directional Derivative, Normal to Level Curve and Tangent Planes, Line Integral, Line Integral of Vector Fields, Conservative Fields, Green's Theorem, Surface Integral of a Vector Field, Stokes' Theorem, Gauss' Divergence Theorem.

Unit 3 Sequences and Series **12**

Sequences - Convergence and Divergence - Series - geometric series, Convergence Tests (Integral Test, Comparison Tests, Ratio Test, Root Test) - Alternating Series - Absolute and Conditional Convergence - Power Series - Taylor and Maclaurin Series.

Unit 4 First-Order Differential Equations **12**

Introduction to Differential Equations - Classification of Differential Equations (Order, Linearity - Solutions of Differential Equations - Initial Value Problems - First-Order ODEs: Separable Equations, Linear Equations, Exact Equations, - Applications of First-Order ODEs.

Unit 5 Functions of Several Variables **12**

Partial differentiation – Homogeneous functions and Euler's theorem – Total derivative – Change of variables – Jacobians – Partial differentiation of implicit functions – Taylor's series for functions of two variables – Applications: Maxima and minima of functions of two variables and Lagrange's method of undetermined multipliers.

TOTAL HOURS: 60

References:

1. Stewart, J. (2015). *Calculus*. Cengage Learning.
2. Thomas, G. B., Weir, M. D., & Hass, J. (2014). *Thomas' Calculus*. Pearson.
3. Zill, D. G., & Wright, W. S. (2017). *Differential Equations with Boundary-Value Problems*. Cengage Learning.
4. Boyce, W. E., DiPrima, R. C., & Meade, D. B. (2017). *Elementary Differential Equations and Boundary Value Problems*. Wiley.

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	MD	**ENGINEERING PHYSICS**	**3**	**0**	**0**	**3**	**3**

OBJECTIVES:

This is an introductory physics course that aims to enhance the understanding of physics concepts in the different areas of speech, ultrasonics, lasers, optical communication and quantum mechanics that are of relevance to respective engineering programs. The course will enable the students to

- Understand the perception of sound, its metrics, and production and applications of ultrasound.
- Understand basics of quantum mechanics and associated theories.
- Explain physics of semiconductors.
- Describe the principle of laser action and working of lasers.
- Analyse the propagation of light through optical fibres and its application to communication systems and sensors

ACOUSTICS: 4

Classification- Music & Noise - Characteristics of Sound: Pitch/Frequency, Loudness/Intensity- decibel scale - Weber–Fechner law – Loudness Curves- Quality/Timbre

ULTRASONICS: 5

Production - Magnetostriction and Piezoelectric methods – Detection – Piezoelectric, Acoustic grating – Non Destructive Testing – pulse echo system -reflection and transmission modes –Modes of data presentation- A, B and C scan displays – Sonogram.

QUANTUM PHYSICS **9**

Planck's theory (derivation) – Deduction of Wien's displacement law and Rayleigh-Jeans law from Planck's

theory – Properties of Matter waves - wave particle duality - Schrödinger's wave equation – Time independent and time dependent equations – Physical significance of wave function – Particle in a one dimensional box and extension to three dimensional box – Degeneracy of electron energy states – Quantum free electron theory – Density of states – Fermi-Dirac statistics– Free electron concentration in metals.

SEMICONDUCTORS **9**

Classification of semiconductors based on doping and band gap – Intrinsic semiconductor – Concept of hole – carrier concentration derivation –Fermi level and its variation with temperature – electrical conductivity – band gap determination –Extrinsic semiconductors – Carrier concentration derivation in n-type and p-type semiconductors – Variation of Fermi level with temperature and impurity concentration.

LASERS **9**

Interaction of Radiation with Matter-Spontaneous and stimulated emissions– Einstein's A and B coefficients – Conditions for Laser action –Population inversion – Active medium – pumping schemes – Optical resonant cavity- Light Amplification-Types of lasers – Nd: YAG, CO_2 and Semiconductor lasers- homo junction & hetero junction laser.

FIBRE OPTICS **9**

Principle and propagation of light in optical fibres –Numerical aperture and Acceptance angle, Classification of optical fibres (material, mode & refractive index) – Losses in fibres – attenuation, dispersion – Fibre optical communication system (Block diagram) – Active and passive sensors – pressure, strain, displacement.

Total hours 45

TEXT BOOKS:

1. M. N. Avadhanulu, P. G. Kshirsagar , "A text book of Engineering Physics" , S. Chand & Co. Ltd. Revised Edition 2014

REFERENCE BOOKS:

1. Kasap, S.O., Principles of Electronic Materials and Devices, (Special Indian Editio McGraw-Hill Education, 3rd Edition, 2017.
2. Kittle Charles, "Introduction to Solid State Physics", Wiley, 2004
3. Gaur R.K. and Gupta S.L. Engineering Physics. Dhanpat Rai publishers, 2012
4. Halliday, D., Resnick, R. & Walker, J. Principles of Physics. Wiley, 2015.
5. James Beauchamp (Editor) - Analysis, Synthesis, and Perception of Musical Sounds –The Sound of Music-Springer; 2007 th Edition (December 19, 2006)
6. Serway, R.A. & Jewett, J.W. Physics for Scientists and Engineers. Cengage Learning, 2010.
7. Tipler, P.A. & Mosca, G. Physics for Scientists and Engineers with Modern Physics', W.H.Freeman, 2007.

OUTCOMES WITH MAXIMUM KNOWLEDGE LEVELS OF ASSESSMENT:

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

	Course Outcome	Knowledge Level
CO1	Analyse the characteristics and perception of sound, understand production and detection of ultrasonic waves and their applications.	K4
CO2	Explain the basic quantum mechanical concepts and their applications.	K3
CO3	Analyse the physics of semiconductors	K4
CO4	Explain the principle, properties and working of different types of lasers	K3
CO5	Illustrate the principle of light propagation, causes for losses and dispersion and evaluate their impact on fiberoptic communication systems and working of fiberoptics displacement, pressure and strain sensors	K4

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	MD	ENGINEERING PHYSICS LABORATORY	0	0	4	2	4

COURSE OBJECTIVE:

- To determine some of the physical, electrical and optical properties of materials.

LIST OF EXPERIMENTS:

1. Determination of velocity and compressibility of the given liquid - Ultrasonic interferometer
2. Determination of Planck's Constant
3. Determination of specific resistance of the given wire - Carey Foster Bridge
4. Determination of energy bandgap of the given semiconductor - Band Gap of Semi-Conductor
5. Determination of grating element / average size of the particles of a given powder sample using laser
6. Determine the numerical aperture, acceptance angle & losses in fibres of the given optical fibre cable.

Advanced Physics Lab (Open-ended)

1. Mini Project

Course Outcome:

On successful completion of this course, the students will be able to:

	Course Outcomes
CO1	Determine material properties of medium associated with ultrasound propagation
CO2	Use lasers for study of materials and optical fibres.
CO3	Determine electrical properties of metals and semiconductors like specific resistance of a conductor, band gap of semiconductors
CO4	Develop appropriate procedures/methodologies based on study objectives

TOTAL HOURS: 30

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	MC	Problem Solving and Programming in C	2	0	4	4	6

COURSE OBJECTIVES:

- To develop computational thinking skills and apply algorithmic approaches to solve problems using C programming language.
- To implement structured programming concepts including control statements, functions, arrays, and strings for modular program development.
- To apply advanced C programming concepts such as pointers, dynamic memory allocation, structures, and file handling in problem solving.
- To develop the ability to implement, test, and debug C programs using proper coding practices and logical thinking for efficient software development.

Unit 1: PROBLEM SOLVING **3**

Computational Thinking – Logical and Algorithmic Thinking - Basic data types - Operators and expressions - Data input and output statements.

Unit 2: CONTROL STATEMENTS AND FUNCTIONS **3**

Control Statements: Decision making statements, Looping constructs - Function prototype - Function definition and call - Passing arguments to a function - Recursion - Standard library functions - Storage classes.

Unit 3: ARRAYS AND STRINGS **3**

Arrays: Declaration, Initialization, Multi-dimensional arrays, Passing arrays to functions, Bubble sort, Insertion sort, Linear and Binary search - Arrays and Strings, String library functions.

Unit 4: POINTERS **3**

Pointer declarations - Passing pointer s to functions - Pointers and one-dimensional arrays - Dynamic memory allocation - Operations on pointers - Array of pointers – Pointer to pointer.

Unit 5: STRUCTURES AND FILES **3**

Structures: User-defined data types (typedef) – Array of structures - Structures and pointers – Passing structures to functions – Files.

LECTURE PERIODS: 15

SUGGESTED EXPERIMENTS

1. **Solve problems by choosing suitable data types and programming constructs.**
   - *Eg. Design a water billing system for the Eureka Water Company. The company charges its customers based on their water usage, following the billing structure as follows:*

- *Flat Rate (Category A): For water usage up to and including 1000 cubic feet, a flat rate of Rs. 750/- is applied.*
- *Tiered Rate (Category B): For water usage over 1000 cubic feet and up to and including 2000 cubic feet, a rate of Rs. 5/- per cubic foot is applied.*
- *Tiered Rate (Category C): For water usage over 2000 cubic feet and up to and including 3000 cubic feet, a rate of Rs. 10/- per cubic foot is applied.*
- *Flat Rate (Category D): For water usage over 3000 cubic feet, an additional flat rate of Rs. 4500/- is applied.*
- *Develop an application that calculates the water bill for a customer based on their water usage in cubic feet.*

2. **Applications using looping constructs.**
   - *Eg. Generating twin-prime sequence, generating first N automorphic numbers.*
3. **Applications using functions with different parameter passing techniques: Call by value, call by reference; Recursive functions**
   - *Eg. Develop a digital countdown timer for a cooking app. The timer shows the remaining time in hours, minutes, and seconds. Sometimes the user may want to either extend the timer (e.g., add 90 seconds), or shorten it (e.g., subtract 75 seconds). Define a helper function called adjust_time that can handle these adjustments.*
4. **Applications using recursive functions.**
   - *Eg. A joker in circus is trying to juggle 5 discs and throw them to arrange it in 3 rods. He is using Towers of Hanoi approach to accomplish the same. Towers of Hanoi is a mathematical puzzle with three rods and 'n' numbers of discs. Only a single disc is allowed to be transferred at a time. Each transfer or move should consist of taking the upper disk from one of the stacks and then placing it on the top of another stack i.e. only a topmost disk on the stack can be moved. A larger disk cannot be placed over a smaller disk; the placing of disk should be in increasing order.*
5. **Applications using one-dimensional, two-dimensional arrays and passing arrays to functions.**
   - *Eg. You are given a 9 x 9 Sudoku board. A Sudoku board is valid only if the following rules are followed: Each row must contain the digits 1-9 without duplicates. Each column must contain digits 1-9 without duplicates.*
6. **Applications using strings and string library functions.**
   - *Eg. Develop a system to analyze DNA sequences. A DNA sequence is a long string of characters representing nucleotides. The four possible nucleotides are represented by the characters 'A', 'C', 'G', and 'T'. You are given a DNA sequence string and a subsequence string, the goal is to find all occurrences of the subsequence in the main DNA sequence, rack the positions where the subsequence occurs, and the system can ignore case when performing matching.*
7. **Applications using pointers**
   - *Eg. You are given an integer array height of length n. There are n vertical lines drawn such that the two endpoints of the ith line are (i, 0) and (i, height[i]). Find two lines that together with the x-axis form a container, such that the container contains the most water. Return the maximum amount of water a container can store. Notice that you may not slant the container.*
8. **Applications to demonstrate structure manipulations and passing structures to a function**
   - *Eg. Design a Bank Account Management System that stores information such as the account holder's name, account number, balance, and account type (savings, current), and performs the following*

operations: create an account, deposit and withdraw money, display account details and balance, perform transactions on the account.

9. Applications to demonstrate file handling techniques

- *Eg. Simulate an employee management system to store, retrieve, and modify employee data stored in a file.*

10. Game Application using random number generator

- *Eg. Develop an application of tic-tac-toe game in which the user plays the game against the computer. The computer can be either the first player or the second player. If the computer is the first player, let the first move be generated randomly. Display the complete status of the game after each move and show the winner of the game.*

LABORATORY PERIODS: 30

Lecture hours (L): 15 Periods
Practical hours (P): 30 Periods
Total hours (TH): 45 Periods

TEXTBOOKS:

1. Bryon Gottfried, Programming with C (Schaum's Outline Series), McGraw-Hill Education, 4th Edition, 2018.
2. Brian W Kernighan and Dennis M Ritchie, The C Programming Language, Pearson Education India, 2nd Edition, 2015.
3. Karl Beecher, ``Computational Thinking – A beginner's Guide to Problem Solving and Programming", British Computer Society (BCS), 2017.

REFERENCE BOOKS

1. Reema Thareja, Programming in C, Oxford University Press, 2nd Edition, 2016.
2. Yashwant Kanetkar, Let Us C, BPB Publications, 14th Edition, 2016.
3. King, K.N., C Programming A Modern Approach, W. W. Norton & Company, II Ed. 2008.
4. Yashwant Kanetkar, Understanding Pointers in C and C++, 2019.
5. Maureen Sprankle, Jim Hubbard, "Problem Solving & Programming Concepts", Pearson Education, 9th edition, 2011.

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Apply logical and algorithmic approaches to solve problems using basic data types and input/output statements in C.	K3
CO2	Make use of decision-making statements, looping constructs, and functions to develop modular programming.	K3
CO3	Analyze and implement applications using arrays and strings by applying sorting, searching, and string manipulation techniques to solve problems.	K4
CO4	Analyze and apply pointers effectively in C using dynamic memory allocation for program development.	K4

CO5	Develop applications using structures and user-defined data types and perform file handling operations for data storage and retrieval.	K4

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	MC	UNIX SCRIPTING ENVIRONMENT	1	0	2	2	3

OBJECTIVES

- To understand the basics of UNIX operating system
- To learn basic UNIX commands and scripts
- To understand the principles of Unix File system
- To learn Shell Programming

UNITI UNIX BASICS **6**

Unix organization and structure - Introduction to flavours of Unix - Unix environment - Types of Shell - Editors: nano and vi - Environment variables - Customizing the prompt.

UNIT II UNIX FILE SYSTEM AND COMMAND UTILITIES **6**

Terminal - File Systems Navigation - Exploring the File Systems - Manipulating Files and Directories - Getting Commands Documentation - File and Permissions - Users and Groups

UNIT III UNIX COMMANDS **6**

Redirection: Output, Input and Errors - Pipelines - Filters; Expansion and Quoting - Command line editing - Searching for files - Archiving and Backup - Regular Expressions- Text Processing (sed)

UNIT IV BASIC SHELL PROGRAMMING **6**

Shell scripts – PATH - Variables and Constants - Shell functions and Local variables- Flow Control: Branching with if & Case - Reading from standard input

UNIT V LOOPING AND PROCESSES **6**

Looping with while/until/for - Positional Parameters - Strings and Numbers: Parameter expansion, Arithmetic Evaluation and Expansion – Arrays - Processes

TOTAL PERIODS: 30

LIST OF EXPERIMENTS

1. Experiment with basic Unix commands.
2. Make use of Unix commands for file and directory handling.
3. Experiment with security and file permissions.
4. Make use of pipes, quotes, aliases and variables.

5. Make use of filters.
6. Experiment with sed
7. Develop Bash shell scripts to search for a pattern in a file
8. Develop Bash shell script to reverse and print the digits of a number.
9. Develop scripts using signals and cron jobs

TOTAL PERIODS (PRACTICAL): 30

TOTAL HOURS :60

COURSE OUTCOMES

On successful completion of this course, the student will be able to:

1. Explain the structure and functionalities of Unix operating System. (K2)
2. Use Unix commands to manage files, directories, and permissions effectively. (K3)
3. Demonstrate the working of Unix file system using command line Utilities. (K3)
4. Develop simple shell scripts using variables, control statements, and loops to automate tasks. (K3)
5. Apply Unix process management and redirection techniques to handle multiple tasks and data streams.(K3)

TEXTBOOKS

1.Behrouz Forouzan, Richard Gilberg "UNIX and Shell Programming: A Textbook", Thomson Learning, 2005 (Unit I).
2. William Shotts A "The Linux Command Line",Fifth Internet Edition (Unit I, Unit II(chapter 2,3,4,5,9), Unit III (Chapter : 6,7,8 17,18,19), Unit IV(Chapter: 24,25,26,27,28,31), Unit V (Chaper:10,29,32,33,34,35)
3. Brian W. Kernighan and Rob Pike, "Unix Programming Environment", AT&T Murray Hills, NewJersy.(Unit II- Chapter 2)

REFERENCES

1. Mike Loukides, Tim O'Reilly, Jerry Peek, Shelley Powers, "Unix Power Tools", 3rd Edition, O'Reilly, 2009
2. Sumitabha Das, "Unix: Concepts and Applications", Tata McGraw Hill,2017.
3. Richard Blum, Christine Bresnahan, "Linux Command Line and Shell Scripting Bible", Wiley, 2015.
4. Stephen Kochan, Patrick Wood, "Shell Programming in UNIX, Linux and OSX",Addison Wesley, 2016
5. Randal K Michael, "Mastering Unix Shell Scripting", 2nd Edition, Wiley, 2008.

Course Code	Category	Course Title	L	T	P	C	Contact Hours
	MC	Digital Design + Lab	2	0	2	3	4

Objectives:

- To provide a strong foundation in number systems and Boolean algebra for understanding digital logic design.
- To develop the ability to simplify and analyze Boolean expressions using systematic methods and implement them using logic gates.
- To enable students to design and analyze combinational and sequential digital circuits using standard design procedures.
- To introduce the concepts of storage elements, registers, counters, and memory organization in digital systems.
- To familiarize students with programmable logic devices and basic error detection and correction techniques used in digital systems.

UNIT I NUMBER SYSTEM AND BOOLEAN ALGEBRA 6

Binary Numbers – Number - Base Conversions – Octal and Hexadecimal Numbers – Complements of Numbers – Signed Binary Numbers – Binary Code; Basic Theorems and Properties of Boolean Algebra – Boolean Functions – Canonical and Standard Forms – Digital Logic Gates.

UNIT II SIMPLIFICATION OF BOOLEAN EXPRESSION 6

Minterm – Maxterm – Sum of Products (SOP) – Product of Sums (POS) – Simplification of Boolean Functions using Karnaugh Map – Four Variable K-Map – Five Variable K -Map Logic Don't-Care Conditions, NAND–NOR implementations.

UNIT III COMBINATIONAL LOGIC 6

Combinational Circuits – Design Procedures – Binary Adder–Subtractor – Decimal Adder – Binary Multiplier – Magnitude Comparator – Decoders – Encoders – Multiplexers, HDL.

UNIT IV SEQUENTIAL LOGIC 6

Sequential Circuits – Storage Elements: Latches, Flip–Flops; Analysis of Clocked Sequential Circuits – State Reduction and Assignment – Design Procedure – Registers, Counters.

UNIT V MEMORY 6

RAM – Memory Decoding – Error Detection and Correction - ROM – PROM - Programmable Logic Array – Programmable Array Logic.

LAB SUGGESTED EXPERIMENTS

1. Verification of Boolean Theorems using basic gates
2. Design and implementation of combinational circuits using basic gates.
3. Design and implement code converters
4. Design and implement 4-bit binary adder/subtractor

5. Design and implement parity generator/checker
6. Design and implement encoder, decoder and priority encoder
7. Design and implement boolean function using multiplexers
8. Design and implement shift-registers
9. Design and implement synchronous counter
10. Design and implement ripple counter

Lecture hours (L): 30 Periods
Practical hours (P): 30 Periods
Total hours (TH): 60 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Explain the fundamental principles of Boolean algebra and relationships between logic gates	K2
CO2	Utilize K-Map to simplifying Boolean function	K3
CO3	Analyze and construct Combinational Circuits	K4
CO4	Analyze and construct Synchronous Sequential Circuits	K4
CO5	Build various memories and Boolean functions using Programmable Logic Devices	K3

TEXTBOOKS:

1. Morris R Mano, Michael D Ciletti, "Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog", 6th Edition, Pearson Education, 2018.

REFERENCE BOOKS:

1. S Salivahanan, S Arivazhagan, "Digital Circuits and Design", 5th edition, Oxford University Press, 2018.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, "Computer Organization and Embedded Systems", 6th Edition, Tata McGraw Hill, 2012.
3. G K Kharate, "Digital Electronics", Oxford University Press, 2010.
4. John F Wakerly, "Digital Design Principles and Practices", 5th Edition, Pearson Education, 2017.
5. Charles H Roth Jr, Larry L Kinney,"Fundamentals of Logic Design", 6th Edition, Cengage Learning, 2013.

SEMESTER II

Course Category	Course Title	L	T	P	C	Contact Hours
AEC	Technical Writing	2	1	0	3	3

OBJECTIVES:

- To prepare effective instructions, checklists and emails
- To write different kinds of paragraphs and argumentative essays
- To develop and revise effective resumes, cover letters and SOPs
- To document and organise various kinds of reports, case studies and white papers
- To create well-structured proposals and press releases

UNIT I WRITING INSTRUCTIONS, CHECKLISTS AND EMAILS 6 + 3

Writing instructions and checklists; using bulleted lists and numbered lists. Writing professional emails: appropriate greetings, subject line, writing three-body paragraphs – introduction, body, and closing paragraph; discussion of samples. Reading sample instructional texts and professional email models.

Language skills: Tenses – present and past

UNIT II WRITING ARGUMENTATIVE PARAGRAPHS AND ESSAYS 6 + 3

Writing paragraphs. Structure of a paragraph: basic, compare and contrast. Product descriptions and process descriptions. Interpreting charts and graphs. Reading and writing model argumentative essays and sample analytical paragraphs; handling three types of arguments: Classical, Rogerian, Toulmin; structure of an argumentative essay: introduction, building the thesis, body paragraphs with three arguments or claims and supporting evidence, conclusion. Sample essays.

Language skills: Connectors and definitions

UNIT III WRITING JOB APPLICATIONS AND SUMMARIES 6 + 3

Writing job applications, resumes, and cover letters. Writing the statement of purpose (SOP):
Outline – professional narrative, previous experience, research interests, and career goals. Writing summaries: reading the text; identifying main and supporting ideas. Writing paraphrases:
step-by-step paraphrasing; text comparison; sentence analysis.

Language skills: subject-verb agreements

UNIT IV WRITING REPORTS, CASE STUDIES AND WHITE PAPERS 6 + 3

Writing datasheets: outline – descriptions, graphics, benefits, features, specifications, requirements, and contact information. Writing abstracts. Reading sample reports, case studies and whitepapers. Writing technical reports; practising IMRAD structure; Engineering reports & Feasibility reports. Writing case studies and white papers: main components of an issue case study – problems, search for solutions, implementation of the solution and results.

Language skills: Active voice, passive voice

UNIT V WRITING PROPOSALS AND PRESS RELEASES 6 + 3

Writing business proposals: format -- title page, table of contents, executive summary, project details, deliverables and

milestones, budget, conclusion. Writing for Websites: Inverted pyramid style; headings and subheading; bulleted lists; short paragraphs with one idea. Writing press releases: outline – headline, date of publication, contact information, summary, intro paragraph, detail paragraphs, and about section.
Language skills: Correction of errors, proofreading and editing

Lecture : 30 Periods
Tutorial : 15 Periods

Total : 45 Periods

COURSE OUTCOMES

At the end of the course, the learners will be able to:

CO1	Write instructions, checklists and emails with clarity and readability
CO2	Compose paragraphs, and structure an argumentative essay effectively
CO3	Create and edit resumes, cover letters and SOPs concisely
C04	Document well-structured reports, case studies and white papers
CO5	Develop effective proposals and press releases

TEXTBOOKS:

1. Laplante, P. A. (2018). Technical writing: A practical guide for engineers, scientists, and nontechnical professionals. CRC Press.
2. David Bonamy. (2013) Technical English Level 2 Course Book. Pearson

REFERENCES:

1. Kmiec, D., & Longo, B. (2017). *The IEEE guide to writing in the engineering and technical fields*. John Wiley & Sons.
2. Lewis Lansford; Peter Astley. (2013). Oxford English for Careers: Engineering 1: Student's Book. Oxford University Press.
3. Mark Ibbotson. (2008). Cambridge English for Engineering, 2008, Cambridge University Press.
4. Rubens, P. (2002). *Science and technical writing: A manual of style*. Routledge.

Course code	Category	Mathematics II	L	T	P	C	Contact Hours
	MC		3	0	0	3	3

Course Outcomes:

Upon successful completion of this course, students will be able to:

1. Apply and analyse probability laws, counting techniques, and Bayes' theorem to solve uncertainty-based problems.
2. Apply and analyse discrete random variables and standard discrete probability distributions to compute probabilities and moments.
3. Apply and evaluate continuous probability distributions, normal distribution, and the Central Limit Theorem in solving real-world problems.
4. Analyse and evaluate relationships between two random variables using joint distributions, correlation, and regression models.
5. Apply, analyse, and evaluate appropriate statistical tests to draw valid inferences from sample data.

Course Objectives:

This course aims to:

1. Apply fundamental probability concepts, counting techniques, and Bayes' theorem to analyse uncertain events.
2. Analyse discrete random variables and apply standard discrete probability distributions to compute moments and probabilities.
3. Evaluate continuous random variables using probability density function, cumulative density function, and continuous distributions including the normal distribution.
4. Analyse joint behaviour of two random variables using marginal, conditional distributions, correlation, and regression techniques.
5. Apply appropriate statistical tests to analyse hypotheses and draw conclusions from sample data.

Syllabus:

Unit 1: Foundations of Probability

Introduction to Probability - Sample Space - Events - Set Operations on Events - Probability Axioms - Conditional Probability - Independent Events - Bayes' Theorem.

Unit 2: Discrete Random Variables and Distributions

Introduction to Random Variables - Discrete Random Variables - Probability Mass Functions (PMF) - Cumulative Distribution Functions (CDF) - Moments - Moment generating functions - Expected Value of Discrete Random Variables - Variance of Discrete Random Variables - Bernoulli Distribution - Binomial Distribution - Poisson Distribution

Unit 3: Continuous Random Variables and Distributions

Continuous Random Variables - Probability Density Functions (PDF) - Cumulative Distribution Function (CDF) for

Continuous Variables - Moments - Moment generating functions - Expected Value and Variance of Continuous Random Variables - Uniform Distribution (Continuous) - Exponential Distribution - Normal Distribution - Central Limit Theorem.

Unit 4: Two-Dimensional Random Variables

Marginal Distributions - Joint Distributions - Conditional Distributions - Covariance - Correlation - Linear Regression

Unit 5: Test of Significance

Sampling Distributions - Large Sample Test - Test based on Normal: Single and Difference of Mean and Proportion - Small Sample Test - t-distribution: Single and Difference of Means - F-test for Variance - $\chi 2$-test - Test for Goodness of Fit, Independence of Attributes.

References:

1. Devore, J. L. (2015). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning.
2. Ross, S. M. (2014). *A First Course in Probability*. Pearson.
3. Montgomery, D. C., & Runger, G. C. (2014). *Applied Statistics and Probability for Engineers*. Wiley.
4. Wackerly, D. D., Mendenhall III, W., & Scheaffer, R. L. (2008). *Mathematical Statistics with Applications*. Cengage Learning.

Course Code	Category	Data Structures	L	T	P	C	Contact Hours
	MC		3	0	0	3	3

Course Objectives:

- To comprehend the notion of linear and non-linear data structures.
- To apply the linear and non-linear data structures to various problems.
- To learn hashing techniques and disjoint set operations.
- To choose a suitable data structure to solve the given problem.

Unit 1: LINEAR DATA STRUCTURES – LIST 9

Abstract Data Type (ADT); List ADT: Array based and Linked List based Implementations; Doubly Linked List; Circular Linked List; Applications of List: Polynomial Manipulation; Asymptotic Notations: big-Oh; Efficiency of algorithms: Notion of time and space complexity; Analysis of non-recursive and recursive algorithms.

Unit 2: LINEAR DATA STRUCTURES – STACKS AND QUEUES 9

Stack ADT: Stack model – Implementation of stacks – Applications: Balancing symbols – Infix to postfix conversion – Evaluating postfix expressions; Queue ADT: Queue model – Implementation of queues – Applications of queues.

Unit 3: NON-LINEAR DATA STRUCTURES – TREES 9

Tree Preliminaries; Binary Tree: Traversal on Trees, Expression Tree; Threaded Binary Tree; Binary Search Tree ADT; AVL Tree operations and implementation; Priority queues using Binary Heap; Tries

Unit 4: NON-LINEAR DATA STRUCTURES – GRAPHS 10

Graphs: Basic definition and Representation of Graphs; Graph Traversals: Breadth First Search and Depth First Search; Topological Sort; Shortest Path: Dijkstra's Algorithm, Floyd Warshall Algorithm.

Unit 5: HASHING AND DISJOINT SET 8

Hash ADT: Hash function, Separate chaining, Open addressing, Rehashing, Extendible hashing; Applications of Hashing technique; Algorithms for Disjoint Set operations.

Lecture hours (L): 45 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Use linear data structures namely arrays and linked lists for developing applications by analysing their time complexity.	K4
CO2	Use Stack and Queue data structures to develop solutions for various problems.	K3
CO3	Solve problems using different types of tree data structures.	K3
CO4	Choose appropriate graph algorithms to solve problems.	K5
CO5	Solve problems applying hashing and using disjoint sets.	K3

TEXTBOOKS:

1. Seymour Lipschutz, "Data Structures with C", Schaum's Outline Series, McGraw Hill Education (India), 2017. (All Units)
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 4th Edition, PHI Learning Pvt. Ltd. 2022.

REFERENCE BOOKS:

1. M A Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, 2002.
2. Byron Gottfried, "Schaum's Outline of Programming with C", 4th edition, McGraw Hill Education (India), 2019.
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekar, "Fundamentals of Computer Algorithms", 2nd Edition, University Press, 2017.
4. S Sridhar, "Design and Analysis of Algorithms", 1st Edition, Oxford University Press, 2015.
5. Richard F Gilberg, Behrouz A Forouzan, "Data Structures: A Pseudocode Approach with C", 2nd Edition, Cengage India, 2007.
6. A V Aho, J E Hopcroft, J D Ullman, "Data Structures and Algorithms", Pearson Education, 1st Edition Reprint, 2003.

Course Code	Category	Computer Organization and Architecture	L	T	P	C	Contact Hours
	MC		3	0	0	3	3

Course Objectives:

- To learn the fundamental building blocks and arithmetic operations in computer systems.
- To understand the MIPS instruction set architecture and MIPS assembly language
- To explore the principles of processor design and implementation with emphasis on non-pipelined and pipelined architectures.
- To understand memory hierarchy and input/output mechanisms including IO modes and DMA.
- To provide insights into parallel architectures and their challenges in modern computing.

UNIT I Basic Structures of Computers & Arithmetic Design 09 hrs

Functional Units: Basic Operational Concepts, Adder and subtraction; Multiplication: Booths and modified Booths; Division : restoring and non-restoring; Floating Point Representation, Floating Point Operations.

UNIT II MIPS Instruction Set Architecture 09 hrs

Performance Metrics and Evaluation, MIPS Operands; MIPS Instructions: Arithmetic instructions, Logical Operations, Instructions for Making Decisions; MIPS assembly Language; MIPS Addressing Modes.

UNIT III Processor Design 10 hrs

Building a Data path: Logic Design Conventions, Simple Implementation Scheme;
MIPS Pipelining: Overview, Pipelined Data path and Control; Data Hazards: Forwarding versus Stalling; Control Hazards, Exceptions; Parallelism via Instructions.

UNIT IV Memory Unit and I/O Modes 09 hrs

Memory Hierarchy, Memory Technologies: SRAM & DRAM; Cache Memory: Basics, Measuring and Improving Cache Performance, Cache Replacement Techniques, Cache Coherence; IO Modes: Programmed IO, Interrupt Driven IO, Direct Memory Access.

UNIT V Parallel Processor Architectures 08 hrs

Parallel processing challenges, Flynn's classification, Vector Architectures, Hardware multithreading, Multi-core processors, Introduction to Graphics Processing Units.

Total hours: 45 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Illustrate the structure and functionality of fundamental computer system components, and **implement** integer and floating-point arithmetic operations..	K3
CO2	**Analyze** and apply the MIPS instruction set architecture to design and implement low-level programs.	K4
CO3	**Design** and construct processor datapaths using both non-pipelined and pipelined architectures..	K3
CO4	**Develop** memory organization and apply techniques to improve memory	K3

	and I/O performance.	
CO5	Illustrate the principles and architectures of parallel processing systems.	K2
CO6	To model, simulate, and validate arithmetic and memory designs by **Utilizing** engineering tools (e.g., MIPS Simulator, Logisim, VHDL/Verilog)	K5

TEXTBOOKS

1. David A Patterson, John L Hennessy, "Computer Organization and Design: The Hardware/Software Interface", 6th Edition, Morgan Kaufmann / Elsevier, 2021. (Unit -1, 2, 3, 4: Memory Part)
2. William Stallings, "Computer Organization and Architecture: Designing for Performance", 11th edition, Pearson Series, 2019. (Unit-4: IO Modules, Programmed IO, Interrupt Driven IO, Direct Memory Access.)
3. John L Hennessey, David A Patterson, "Architecture – A Quantitative Approach", 7th edition, Morgan Kaufmann, Elsevier, 2025 (Unit-5)

REFERENCE BOOKS:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, "Computer Organization and Embedded Systems", 6th Edition, Tata McGraw Hill, 2022.
2. Morris Mano M, "Computer System Architecture", Revised 3rd Edition, Pearson Publication, 2017
3. John P Hayes, "Computer Architecture and Organization", 3rd Edition, Tata McGraw Hill, 2012.

Course Code	Category	Object Oriented Programming	L	T	P	C	Contact Hours
	MC		2	0	2	3	4

Course Objectives:

- To learn the basic concepts of object-oriented programming using C++.
- To learn how to design programs using classes, objects, inheritance and polymorphism.
- To develop programs using exception handling and file handling techniques.
- To apply templates and Standard Template Library components in problem solving.
- To use namespaces and lambda expressions for writing modular and efficient C++ programs.

UNIT I DATA ABSTRACTION & OVERLOADING 6

Overview of C++ – Structures – Class Scope and Accessing Class Members – Reference Variables – Initialization – Constructors – Destructors – Member Functions and Classes – Friend Function –Dynamic Memory Allocation – Static Class Members - Overloading: Function overloading and Operator Overloading.

UNIT II INHERITANCE AND POLYMORPHISM 6

Base Classes and Derived Classes – Protected Members – Casting Class pointers and Member Functions – Overriding – Public, Protected and Private Inheritance – Constructors and Destructors in derived Classes – Implicit Derived – Composition Vs. Inheritance – Virtual functions – This Pointer – Abstract Base Classes and Concrete Classes – Virtual Destructors – Dynamic Binding.

UNIT III EXCEPTION HANDLING AND FILES 6

Exception Handling: Benefits of exception handling - Exception hierarchy - Usage of try, catch, throw, rethrowing exceptions - Built in exceptions - Creating own exception sub classes; Files – Operations - File streams – Random access file- Exception Handling Files.

UNIT IV TEMPLATES 6

Function templates – Class templates – Standard Template Library: Containers: Sequence containers – Associate containers – Iterators – Algorithms

UNIT V NAME SPACES AND LAMDA EXPRESSIONS 6

Namespaces: Using Directives – Declaring Namespace – Global Namespace – Nested Namespace – Inline Namespace - Basic Lambda Expressions - Capture Clause: Mutable lambdas – Generic lambdas – Usage with STL – Recursive lambda expressions

SUGGESTED EXPERIMENTS

1. Programs Using Functions - Functions with Default Arguments, Implementation of Call by Value- Call by Address
2. Simple Classes for understanding objects- member functions and Constructors – Friend function
3. Compile Time Polymorphism - Operator Overloading including Unary and Binary Operators, Function Overloading
4. Runtime Polymorphism – Inheritance, Virtual Functions, Virtual Base Classes
5. Programs using **Exception Handling** – Demonstrate `try`, `catch`, `throw` and rethrowing exceptions. Handle errors such as division by zero and invalid input.
6. Programs using **User Defined Exceptions** – Create custom exception classes and demonstrate the exception hierarchy. Handle different types of exceptions using multiple `catch` blocks.
7. Programs using **Random Access Files** – Store multiple records in a file and retrieve specific records using file positioning functions like `seekg()` and `seekp()`.
8. Programs using function and class templates
9. Programs using **Lambda Expressions with STL** – Use lambda expressions with STL algorithms for sorting, searching and filtering elements in containers.

Lecture hours (L): 30 Periods
Practical hours (P): 30 Periods
Total hours (TH): 60 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Implement C++ programs using classes, objects, constructors, destructors and operator overloading to achieve data abstraction.	K3
CO2	Analyze and implement inheritance and polymorphism concepts such as virtual functions, abstract classes and dynamic binding in object-oriented design.	K4
CO3	Develop programs using exception handling mechanisms and file handling techniques for reliable and persistent data processing.	K3
CO4	Apply templates and Standard Template Library components such as containers, iterators and algorithms to solve programming problems efficiently.	K3
CO5	Apply namespaces and lambda expressions to design modular and modern C++ programs integrated with STL.	K3

TEXTBOOKS:

1. Timothy A. Budd**,** An Introduction to Object-Oriented Programming, Third Edition, Addison-Wesley, 2001
2. Paul Deitel, Harvey Deitel, C++ How to Program, Eleventh Edition, Pearson Education, 2024.

REFERENCE BOOKS:

1. Robert Lafore, Object-Oriented Programming in C++, Fourth Edition, Addison-Wesley Professional, 2002.
2. Richard Grimes, Beginning C++ Programming, Packt Publishing, 2017
3. Bjarne Stroustrup, "The C++ Programming Language", 3rd Edition, Pearson Education, 2007.
4. Bhushan Trivedi, "Programming with ANSI C++, A Step-By-Step approach", Oxford University Press, 2010.
5. Ulla Kirch-Prinz, Peter Prinz, A Complete Guide to Programming in C++, Jones And Bartlett Publishers, 2002

Course Code	Category	ENGINEERING PRACTICES LABORATORY (Common to CSE/IT/ECE/BME)	L	T	P	C	Contact Hours
	SEC		0	0	2	1	2

Objectives:

- To provide hands-on experience to the students in various basic engineering practices in Mechanical and Civil Engineering
- To provide hands-on training to the students in electrical wiring practices and the use of electronic components.

LIST OF EXPERIMENTS:

Mechanical & Civil Engineering Practice

1. **Power Tool Carpentry – Utility Product Fabrication:** Cutting, drilling, sanding, joint preparation. *Fabrication:* **Pen stand/utility box,** *Outcome:* Confidence in real carpentry operations.
2. **Carpentry Joints & Building Components:** Fabrication using lap and mortise–tenon joints. *Fabrication:* **Mini door/window frame or wall shelf,** *Outcome:* Understanding structural wood joints used on site.
3. **Plumbing Practice – Domestic Water Supply:** Pipe cutting, joining, and fitting (PVC & GI). *Fabrication:* **Wash basin plumbing system,** *Outcome:* Knowledge of household plumbing installation.
4. **Machining Practice – Functional Metal Component:** Drilling, tapping, countersinking, and work holding. *Fabrication:* **Fixture plate/tool holder,** *Outcome:* Exposure to workshop machining operations.
5. **Sheet Metal Fabrication – Industrial Product:** Cutting, bending, folding, riveting. *Fabrication:* **Mobile stand / small enclosure/duct prototype:** *Outcome:* Sheet metal fabrication skills.
6. **Real-Life Mechanical Systems Study:** Hands-on observation and disassembly. *Systems:* **Hand pump, flush tank, pedestal fan mechanism.** *Outcome:* Linking theory with real engineering products.

Electrical & Electronics Engineering Practice

1. **Domestic Electrical Wiring Practice:** Assemble a single-phase domestic wiring circuit for operating household appliances using fuse, switch, lamp, energy meter and test its operation. *Outcome***:** Demonstrate domestic electrical wiring practices.
2. **Electrical Measurements in RLC Circuit**: Assemble a single-phase RLC circuit and measure electrical quantities-voltage, current, power, and power factor. *Outcome*: Understand the practical measurement of electrical quantities in RLC circuit.
3. **Soldering Practice**: Assembly and testing of mobile phone charging circuit. *Outcome:* Hands-on skill in soldering, component identification, and testing.
4. **Relay Control Circuit Using Logic Gates:** Assembly and testing of a relay-based control circuit using logic gates.

Outcome: Understand the operation of a relay control circuit using logic gates.

5. **AC signal Parameters Measurement and Analysis**: Measurement of amplitude and frequency in a rectifier circuit using a CRO. *Outcome:* Understand the usage of CRO and measurement of AC signal parameters.

TOTAL PERIODS 30

COURSE OUTCOMES

On successful completion of this course, students will be able to

CO1: To perform carpentry, plumbing, machining, and sheet metal work to fabricate real-time civil engineering components, domestic water supply systems and to manufacture functional mechanical components.

CO2: Construct electrical circuits and demonstrate their working.

CO3: Demonstrate the implementation of electronic circuits for practical applications.

REFERENCE BOOKS

1. S Gowri, T Jeyapoovan, "Engineering Practices Laboratory Manual", 5th Edition, Vikas Publishing, 2016.
2. V Ramesh Babu, "Engineering Practices Laboratory", VRB Publications, 2006.
3. Willis H Wagner, Howard Bud Smith, Mark W Huth, "Modern Carpentry", 12th Edition, 2015.
4. P C Sharma, "Production Technology: Manufacturing Process", S Chand publisher, 2006.
5. David W Rongey, "A Complete Guide to Home Electrical Wiring", 2013.
6. K Jeyachandran, S Natarajan, S Balasubramanian, "A Primer on Engineering Practices Laboratory, Anuradha Publications, 2007.

Course Code	Category	BASIC ENGINEERING GRAPHICS (Common to CSE/IT/ECE/BME)	L	T	P	C	Contact Hours
	MC		0	0	2	1	2

Objectives:

- To introduce the use of **drawing instruments** for professional practice.
- To enable clear and precise **communication of engineering ideas** through standard drawings
- To establish a fundamental basis for **design and drafting** for engineering applications.

Module 1: Introduction to Engineering Graphics

Importance of engineering drawing in engineering practice, BIS standards for engineering drawing, Drawing instruments and basic construction techniques, Lines, lettering and dimensioning. Geometric constructions and polygons.

Module 2: Conic Sections

Curves used in engineering practices, Construction of ellipse, parabola and hyperbola by eccentricity method. Drawing of tangent and normal to the above curves. Application of conics in engineering and real-life problems.

Module 3: Free Hand Sketches

Principles of orthographic projection, Visualization and conversion from pictorial to orthographic views, Free hand sketching of front, top, and side views from pictorial views of an object.

Module 4: Projection of Points and Lines

Projections of points when they are situated in different quadrants, Projections of a line parallel to one of the reference planes and inclined to the other, line inclined to both the reference planes**.**

Module 5: Projection of Planes

Projection of planes (polygonal and circular surfaces) perpendicular to one of the reference planes and inclined to the other. Planes inclined to both the principal planes.

Course Outcomes:

Upon successful completion of the course, students will be able to

CO1: Interpret and construct the ellipse, parabola and hyperbola by eccentricity method.

CO2: Develop the orthographic views of an object,

CO3: Understand and Construct the Projection of points and lines.

Textbooks:

1. Venugopal, K. and Prabhu Raja, V., Engineering Drawing + AutoCAD, New Age International (P) Limited, 2022, ISBN - 13: 9788122472752.
2. Natarajan, K.V., A Textbook of Engineering Graphics, Dhanalakshmi Publishers, Chennai, 33rd Edition, 2020. [ISBN: 9788190414089].
3. Ramesh babu. V, A Textbook on Engineering Graphics, VRB Publishers, Chennai, 2022. [ISBN - 13: 978-93-95018-55-5.]

References:

1. Bhatt, N.D., Engineering Drawing, Charotar Publishing House, 2025. [ISBN: 9789385039805].
2. Prof. Sham Tickoo, AutoCAD 2017 for Engineers & Designers, 23ed, Dreamtech Press[ISBN: 9789351199465]
3. Basant Agarwal, and Agarwal, C.M., Engineering Drawing, McGraw Hill, 3rd Edition, 2019. [ISBN: 9789353167448]

4. Parthasarathy, N. S. and Vela Murali, “Engineering Drawing”, Oxford University Press,2015.[ISBN: 9780199455393]

Course Code	Category	Data Structures Laboratory	L	T	P	C	Contact Hours
	MC		0	0	4	2	4

Course Objectives:

- To implement linear data structures such as arrays, lists, stacks, and queues using programming techniques to solve computational problems.
- To design and develop non-linear data structures including trees, graphs, hash tables, and disjoint sets in solving complex engineering problems.
- To analyze problem requirements and select appropriate data structures for efficient program implementation.

SYLLABUS

SUGGESTED EXPERIMENTS

1. Applications of Array (e.g. Finding pairs in an array with the given constraints)
2. Applications of Linked List (e.g. Polynomial manipulations)
3. Applications of Doubly Linked List (e.g. Simulation of a music player)
4. Applications of Stack (e.g. Simulation of Undo & Redo operations in a text editor)
5. Applications of Queue (e.g. Simulating printer jobs)
6. Implementation of Expression Tree (e.g. Elimination of common subexpression)
7. Applications of Binary Search Tree (e.g. Finding k^{th} largest and k^{th} smallest element)
8. Applications of Graph Traversal Techniques (e.g. Simulation of network routing protocols)
9. Implementation of Shortest path algorithms (e.g. Finding the shortest route to connect one city to another)
10. Implementation of Hash Table (e.g. Symbol tables in Compilers)
11. Applications of Disjoint Set (e.g. Finding connected components and cycles)
12. Mini Project – Solving classical data structures problems referring to competitive programming websites

Practical hours (P): 60 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Solve the applications using array, list, stack and queue data structures.	K3
CO2	Use trees, graphs, hash tables and disjoint set data structures to solve computational problems.	K3
CO3	Select suitable data structures for solving complex engineering problems.	K5

Course Code	Category	Version Control and Collaboration Lab	L	T	P	C	Contact Hours
	MC		0	0	4	2	4

COURSE OBJECTIVES

- To use version control software to create and manage repositories
- To track and explore changes in a repository
- To manage branches of a repository and resolve conflicts
- To collaborate using a common remote repository

- To use version control for computational work

SUGGESTED EXPERIMENTS

1. Installing and setting up git
   - Navigate the file system using CLI (Command Line Interface)
   - Use command line history and completion
   - Install git
   - Configure git
2. Creating repository, tracking changes
   - Create a repository, .git folder
   - Modify-add-commit cycle
   - Write documents in md (markdown) using a text editor
   - Record notes using md
3. Exploring history
   - Compare versions of tracked files
   - Restore old versions
   - Ignore files that need not be tracked
4. Branches
   - Create branches
   - Switch branches
   - Merge branches
   - Delete branches
5. Remotes in GitHub/GitLab
   - Create remote repostories
   - Connect local to remote repository
   - ssh background and setup
   - Push to a remote repository
6. Conflicts
   - Merge conflicts
   - Resolve conflicts
7. Collaboration
   - Clone a remote repository
   - Push to a common remote repository
   - Pull or fetch from a remote repository
8. Workflow for collaboration
   - Pull request
   - Fork
   - Blame
9. Tags and releases, issues
   - Tags workflow
   - Open, fill, submit issues
10. Rewriting history
    - Accidental commits
    - Remove changes or files
    - Purge a file from a repository's history
11. Open Science
    - Version control as a lab notebook for computational work

TOTAL PERIODS:
Practical hours (P): 60 Periods

COURSE OUTCOMES

	Course Outcomes	Knowledge Level
CO1	Create and configure repositories	K3
CO2	Track changes and explore history of repositories	K4
CO3	Manage (create, switch, merge) branches and resolve conflicts	K4
CO4	Collaborate on a common remote repository: clone, push, pull, fetch, fork	K4
CO5	Use tags, manage issues, revisit changes and use version control for computational work	K4