R 26

B. Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE)

CSE
(ARTIFICIAL INTELLIGENCE)

R26 Regulation

B. TECH I & II Year
COURSE STRUCTURE
AND SYLLABUS

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

INSTITUTE VISION

To emerge as one of the premier institutions through excellence in education and research, producing globally competent and ethically strong professionals and entrepreneurs.

INSTITUTE MISSION

M1: Imparting high-quality technical and management education through the state-of-the- art resources.

M2: Creating an eco-system to conduct independent and collaborative research for the betterment of the society

M3: Promoting entrepreneurial skills and inculcating ethics for the socio-economic development of the nation.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT VISION

To impart quality education and research in Computer Science and Engineering for producing technically competent and ethically strong IT professionals with contemporary knowledge

DEPARTMENT MISSION

M1: Achieving academic excellence in computer science through effective pedagogy, modern curriculum and state-of-art computing facilities.

M2: Encouraging innovative research in Computer Science and Engineering by collaborating with Industry and Premier Institutions to serve the nation.

M3: Empowering the students by inculcating professional behavior, strong ethical values and leadership abilities

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: To provide software solutions for arising problems in diverse areas with strong knowledge in innovative technologies of Computer Science

PEO2: To serve as globally competent computer professionals and entrepreneurs or in pursuit of higher education and research, developing innovative solutions in multidisciplinary domains.

PEO3: To attain the professional etiquette, soft skills, leadership, teamwork, ethical values in computer science with a commitment for lifelong learning to serve for the society and environment

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: **Analysis & Design:**

Ability to design, develop and deploy customized applications in all applicable domains using various algorithms and programming languages.

PSO2: **Computational Logic:**

Ability to visualize and configure computational need in terms of hardware and software to provide solutions for various complex applications.

PSO3: **Software Development:**

Ability to apply standard procedures, tools and strategies for software development.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE (ARTIFICIAL INTELLIGENCE)

INDUCTION PROGRAMME

S.No.	Course Name	Category	L-T-P-C
1	Physical Activities -- Sports, Yoga and Meditation, Plantation	MC	0-0-6-0
2	Career Counselling	MC	2-0-2-0
3	Orientation To All Branches -- Career Options, Tools, etc.	MC	3-0-0-0
4	Orientation On Admitted Branch -- Corresponding Labs, Tools and Platforms	EC	2-0-3-0
5	Proficiency Modules & Productivity Tools	ES	2-1-2-0
6	Assessment On Basic Aptitude and Mathematical Skills	MC	2-0-3-0
7	Remedial Training in Foundation Courses	MC	2-1-2-0
8	Human Values & Professional Ethics	MC	3-0-0-0
9	Communication Skills -- Focus on Listening, Speaking, Reading, Writing Skills	BS	2-1-2-0
10	Concepts of Programming	ES	2-0-2-0

I B. Tech. – I Semester (CAI)

S.No.	Course Code	Subject	L/D	T	Pr	P	Credits
1	26HS0849	Physics for Advanced Computing	3	0	0	0	3
2	26HS0866	Linear Algebra & Calculus	3	0	2	0	4
3	26HS0821	Communicative English	2	0	0	0	2
4	26CS0501	Computational Thinking and Problem Solving with Python	3	0	0	0	3
5	26CS3101	AI Tools and Applications	1	0	4	0	3
6	26CS0502	Computational Thinking and Problem Solving with Python Lab	0	0	0	3	1.5
7	26HS0850	Physics for Advanced Computing Lab	0	0	0	3	1.5
8	26HS0822	Communicative English Lab	0	0	0	2	1
9	26HS0823	NSS/NCC/Scouts & Guides/Community Service	0	0	0	1	0.5
10	26CS4801	Cybersecurity Awareness for Engineers	2	0	0	0	0
Total			**14**	**0**	**6**	**06**	**19.5**

I B. Tech. – II Semester (CAI)

S.No.	Course Code	Subject	L/D	T	Pr	P	Credits
1	26HS0809	Applied Chemistry for Computer Science	3	0	0	0	3
2	26HS0867	Differential Equations and Vector Calculus	3	0	2	0	4
3	26ME0303	Design Thinking	1	0	4	0	3
4	26CS0503	Data Structures &Efficient Problem Solving	3	0	0	0	3
5	26ME0301	Engineering Graphics	1	0	4	0	3
6	26ME0302	Engineering Workshop	0	0	0	3	1.5
7	26HS0810	Applied chemistry for Advanced Computing Lab	0	0	0	3	1.5
8	26CS0504	Data Structures & Efficient Problem Solving Lab	0	0	0	2	1
9	26HS0824	Health and wellness, Yoga and Sports	0	0	0	1	0.5
Total			**11**	**0**	**10**	**9**	**20.5**

II B. Tech. – I Semester (CAI)

S.No.	Course Code	Subject	L/D	T	Pr	P	Credits
1	26HS0869	Probability and Statistics	3	0	0	0	3
2	26HS0811	Biology for Engineers	2	0	0	0	2
3	26CS0505	Digital Logic and Computer Organization	3	0	0	0	3
4	26CS0506	**Professional Core - II** Algorithms Design and Analysis	3	0	0	0	3
5	26CS0507	**Professional Core - III** Object Oriented Design and Programming	3	0	0	0	3
6	26CS0508	Digital Logic and Computer Organization Lab	0	0	0	2	1
7	26CS0509	**Professional Core - II Lab** Algorithms Design and Analysis Lab	0	0	0	3	1.5
8	26CS0510	**Professional Core - III Lab** Object Oriented Design and Programming Lab	0	0	0	3	1.5
9	26CE0107	Sustainability and Green Engineering	2	0	0	0	2
10	26CS4770	**Skill Enhancement Course I:** AI Assisted Application Development	1	0	0	2	2
11	26HS0826	Technical Paper Writing & IPR	2	0	0	0	0
Total			**19**	**0**	**0**	**10**	**22**

II B. Tech. – II Semester (CAI)

S.No.	Course Code	Subject	L/D	T	Pr	P	Credits
1	26ME0305	Managerial Economics and Financial Analysis	2	0	0	0	2
2	26HS0825	Universal Human Values – Understanding Harmony and Ethical Human Conduct	3	0	0	0	3
3	26CS3103	**Professional Core - IV** Intelligent Data Processing and System Integration	3	0	0	0	3
4	26CS0512	**Professional Core - V** Intelligent Database Management Systems	3	0	0	0	3
5	26CS3104	**Professional Core - VI** Computer Architecture For Intelligent Systems	2	0	2	0	3
6	26CS3102	**Professional Core - VII** Fundamentals of Artificial Intelligence	2	0	2	0	3
7	26CS3105	**Professional Core - IV Lab** Intelligent Data Processing and System Integration Lab	0	0	0	3	1.5
8	26CS0516	**Professional Core - V Lab** Intelligent Database Management Systems Lab	0	0	0	3	1.5
9	26CS4771	**Skill Enhancement Course II:** Data Analytics Using Python	1	0	0	2	2
10	26HS0812	Environmental Science	2	0	0	0	0
Total			**18**	**0**	**4**	**8**	**22**
Mandatory Community Service Project Internship of 08 weeks duration during summer vacation							

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26HS0849) PHYSICS FOR ADVANCED COMPUTING
(Common to CSE & Allied branches)

Prerequisites: Engineering Mathematics, Basic Electronics, Computer Science Fundamentals

Course Objectives:

The objectives of the course is

1. *To equip students with the concepts of interference, diffraction and polarization and their applications in optics and communication systems with basic AI-assisted analysis.*
2. *To introduce lasers, fiber optics and advanced sensing technologies along with simple AI-assisted signal processing techniques.*
3. *To provide basic knowledge of quantum mechanics and quantum computing with simple AI-assisted quantum applications.*
4. *To familiarize students with semiconductors and magnetic materials and their applications using basic AI-assisted material analysis methods.*
5. *To impart knowledge on low temperature physics and superconductivity and their applications with simple AI-assisted monitoring techniques.*

UNIT I	Wave Optics and Quantum Information Processing

Core Content:

Interference: Principle of superposition and coherence; Interference; Types of interference; Interference in thin films (reflection geometry) and Newton's rings. Applications: Determination of wavelength and refractive index. **Diffraction:** Introduction to diffraction; Fresnel and Fraunhofer diffraction; Fraunhofer diffraction due to single, double and N-slits (Diffraction Grating); Dispersive power and Resolving power of grating. Applications: CD/DVD data storage, Barcode scanners. **Polarization:** Types of polarization; Nicol's prism; Half-wave and Quarter-wave plates.

AI Integration:

AI-assisted polarization management for quantum key distribution. AI-integrated holographic interferometry for surface flatness analysis. Free/Open-Source.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib and scikit-learn).

UNIT II	Quantum Photonics and Advanced Sensing

Core Content:
Lasers: Interaction of radiation with matter; Requisites and conditions for laser action; He-Ne laser; Nd-YAG laser; LiDAR basics. Laser-based data communication, Free-space optical communication, Optical data storage, Photonic computing basics, Applications of lasers. **Fiber Optics:** Propagation mechanism; Numerical aperture; Acceptance angle; Types of optical fibres; Classifications and Modes of optical fibres; Attenuation and Losses in optical fibers; V-Number; Distributed Fiber Optic Sensing (DFOS) with AI-Powered Signal Analysis.
AI Integration:
AI-powered signal analysis for distributed fiber optic sensing. Python simulation of laser propagation and fiber optic signal attenuation. Free/Open-Source.
Free/Open-Source AI Tools:
Google Colab, Python, NumPy, SciPy and Scikit-learn/TensorFlow.

UNIT III	Introduction to Quantum Mechanics and Quantum Computing

Core Content:
Introduction to quantum mechanics: de-Broglie concept (matter waves); Heisenberg's Uncertainty Principle; Wave function; Postulates of Quantum Mechanics; Schrödinger's time-independent wave equation; Particle in a one-dimensional potential well; Quantum tunnelling (basic idea). **Introduction to Quantum Computing**: Classical bits vs Qubits; Concept of superposition and Entanglement; Basic Quantum Operations: Hadamard gate (idea of superposition), Pauli-X gate; Basic idea of quantum measurement**. Applications (qualitative only):** AI-Assisted Tunnel Diodes and cold emission of electrons; Basic idea of quantum cryptography and secure communication.
AI Integration:
AI-Assisted Analysis of Tunnel Diode Characteristics using Quantum Tunnelling Concepts. AI-Assisted Quantum Key Distribution for Secure Communication. Free/Open-Source.
Free/Open-Source AI Tools:
Google Colab, Python and Scikit-learn.

UNIT IV	Semiconductors and Magnetic Materials

Core Content:
Semiconductors: Introduction; Fermi energy and Fermi level in intrinsic and extrinsic semiconductors; Expression for concentration of electrons in conduction band and holes in valence band; Electrical conductivity; Expressions for drift and diffusion currents; Hall effect; Expression for Hall coefficient and its applications. **Magnetism:** Origin of magnetism; Dia, Para, Ferro, Ferri and Anti-ferro magnetic materials; Hysteresis; Soft and Hard magnetic materials. Applications (qualitative only): Magnetic data storage (MRAM), Hard Disk Drive and Magnetic shielding.
AI Integration:
AI-Assisted Hall Effect Analysis for Semiconductor Characterization. AI-Assisted Magnetic Data Storage and MRAM Optimisation.

Free/Open-Source AI Tools:
Google Colab, Python and Scikit-learn.

UNIT V	**Low Temperature Physics and Superconductivity**

Core Content:
Introduction: Low temperature and its importance; Properties of materials at low temperatures; Liquefaction of gases (basic concept only); Liquid nitrogen and liquid helium (basic properties and uses). Applications (basic ideas): MRI, Rocket fuels, Cryogenic Cooling of High-Performance Computing System. **Superconductors:** Definitions; Meissner effect; Type I and Type II superconductors; BCS theory: Cooper pairs and energy gap (qualitative); DC and AC Josephson effect. Applications (qualitative only): SQUIDs, magnetic levitation and lossless power transmission. Superconducting qubits: Charge, Flux, Phase.
AI Integration:
AI-Assisted Cryogenic Cooling Optimization for High-Performance Computing Systems. AI-Assisted Stability Control in Superconducting Magnetic Levitation Systems.
Free/Open-Source AI Tools:
Google Colab, Python and Scikit-learn.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply the principles of interference, diffraction and polarization to analyze optical systems and engineering applications.	**L3 (Apply)**
CO2	Analyze the working of lasers, fiber optics and sensing systems with basic AI-assisted optical communication concepts.	**L4 (Analyze)**
CO3	Apply the concepts of quantum mechanics and quantum computing to explain modern computational and secure communication systems.	**L3 (Apply)**
CO4	Analyse semiconductor characteristics and magnetic material behaviour for engineering and electronic applications.	**L4 (Analyze)**
CO5	Apply the concepts of low-temperature physics and superconductivity in modern technological applications and intelligent systems.	**L3 (Apply)**

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	BL	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1		3	2	1	1	2	-	-	-	-	-	-	1
CO2		3	3	2	2	2	-	-	-	-	-	-	1
CO3		3	2	2	2	2	-	-	-	-	-	-	1
CO4		3	3	2	2	2	-	-	-	-	-	-	1
CO5		3	2	1	2	2	-	-	-	-	-	-	1

Textbooks

1.Avadhanulu, M. N., Kshirsagar, P. G., & Arun Murthy, T. V. S. (2019). Engineering Physics. S. Chand.
2.Bhattacharya, D. K., & Tandon, P. (2015). Engineering Physics. Oxford Press.

Reference Books

1.Pandey, B. K., & Chaturvedi, S. (2021). Engineering Physics. Cengage Learning.
2.Pillai, S. O. (n.d.). Solid State Physics. New Age International.
3.Nielsen, M. A., & Chuang, I. L. (n.d.). Quantum Computation & Quantum Information. Cambridge University Press.
4.Saleh, B. E. A., & Teich, M. C. (n.d.). Fundamentals of Photonics. Wiley.
5.Keiser, G. (n.d.). Optical Fiber Communications. McGraw Hill.
6.Tinkham, M. (n.d.). Introduction to Superconductivity. Dover Publications.

Online Resources

1.NPTEL – nptel.ac.in
2.MIT OpenCourseWare – ocw.mit.edu
3.HyperPhysics – hyperphysics.phy-astr.gsu.edu
4.PhET Simulations – phet.colorado.edu
5.IBM Qiskit (free, open-source) – For quantum computing and superconducting qubit simulations on Google Colab

Recommended Free & Open-Source AI/Computational Tools

- Python (NumPy, SciPy, OpenCV, Matplotlib) – FREE on Google Colab for wave optics and signal analysis
- IBM Qiskit (free, open-source) – For quantum computing and superconducting qubit simulations on Google Colab
- TensorFlow / Keras (free, open-source) – For photonic and optical pattern recognition
- scikit-learn (free, open-source) – For semiconductor property prediction and LiDAR signal classification
- PhET Interactive Simulations (free) – For wave optics, quantum mechanics and semiconductors

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	2	0	4

(26HS0866) LINEAR ALGEBRA AND CALCULUS
(Common to all branches)

Prerequisites: Higher Secondary Mathematics

Course Objectives

The objectives of the course is to

1. *Build strong foundations in linear algebra and multivariable calculus with direct engineering relevance.*
2. *Integrate analytical methods with AI-assisted computational tools for visualization, simulation, and verification.*
3. *Develop outcome-oriented problem-solving skills aligned with NEP 2020 and modern engineering applications.*
4. *Apply matrix methods, eigen analysis, and multivariable calculus to solve engineering problems.*
5. *Use computational tools for approximation, optimization, differentiation, and integration.*

UNIT – 1	**Matrix Algebra and Linear Systems**
Content	Matrices, types of matrices, operations, transpose, inverse, symmetric and skew-symmetric matrices; determinant properties; rank of a matrix; echelon and normal forms; system of linear equations; consistency conditions; Gauss elimination and Gauss-Jordan methods; iterative methods: Jacobi and Gauss-Seidel; Cramer's rule; matrix inversion method; applications to engineering systems.
AI Integration	Python-based solution of large linear systems using NumPy; visualization of convergence in iterative methods; AI-assisted debugging of matrix algorithms; using ChatGPT/Copilot to explain row operations and rank reduction.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy for symbolic matrix operations; Jupyter Notebook; Scilab.

UNIT – 2	**Eigenvalues, Eigenvectors, Diagonalization and Quadratic Forms**
Content	Eigenvalues and eigenvectors; characteristic equation; Cayley-Hamilton theorem(without proof); diagonalization; similarity transformation; powers and inverse of matrices using Cayley-Hamilton theorem; quadratic forms;

	canonical form; reduction by orthogonal transformation; rank, index, signature; positive definite forms; orthogonal, Hermitian, skew-Hermitian and unitary matrices.
AI Integration	AI-assisted eigenvalue computation and verification; Python/SciPy visualization of eigenmodes and quadratic surfaces; use of AI tools to explain diagonalization and matrix classification; numerical checking of Cayley-Hamilton theorem.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib); SymPy; GNU Octave; Google Colab.

UNIT – 3	**Single Variable Calculus and Series Expansions**
Content	Rolle's theorem(without proof); Lagrange's mean value theorem(without proof); Cauchy's mean value theorem(without proof); Taylor's and Maclaurin's series with remainder(without proof); approximation of functions; error estimation; curvature of plane curves; radius and centre of curvature; circle of curvature; applications to engineering approximations and curve design.
AI Integration	SymPy-based Taylor and Maclaurin expansion generation; plotting truncation error and convergence; AI-supported derivation checking for mean value theorems; curvature plot visualization and interpretation.
Free/Open-Source AI Tools	Python (SymPy, NumPy, Matplotlib); Google Colab; Jupyter Notebook; Wolfram Alpha free tier for verification

UNIT – 4	**Multivariable Differentiation and Optimization**
Content	Functions of several variables; limits and continuity; partial derivatives; higher order partial derivatives; total derivative; chain rule; tangent plane and normal line; Jacobians; functional dependence; Taylor expansion of two-variable functions; maxima and minima; second derivative test; Lagrange multipliers and constrained optimization; divergence and curl.
AI Integration	Gradient field visualization using Python; AI-assisted Jacobian computation and verification; constrained optimization using SciPy; use of AI tools to interpret saddle points, extrema, and optimization constraints.
Free/Open-Source AI Tools:	Python (NumPy, SciPy, SymPy, Matplotlib); Google Colab; Jupyter Notebook; SciPy optimize module

UNIT – 5	Multiple Integrals and Coordinate Transformations	
Content	Double integrals; evaluation over type I and type II regions; change of order of integration; triple integrals; change of variables; polar, cylindrical and spherical coordinates; area and volume computation; mass, centroid and moment calculations; engineering applications of multiple integration.	
AI Integration	Numerical evaluation of double and triple integrals using SciPy; 2D and 3D region visualization; AI-assisted generation of integration code; interpreting Jacobian as geometric scaling factor in transformed coordinates.	
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib); Google Colab; SymPy; Jupyter Notebook; 3D plotting tools.	

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply matrix algebra techniques to solve systems of linear equations and analyze matrix properties relevant to engineering models.	**L3 (Apply)**
CO2	Analyze eigenvalue problems, diagonalization, and quadratic forms to interpret linear transformations and engineering systems.	**L4 (Analyze)**
CO3	Apply mean value theorems, Taylor and Maclaurin series, and curvature concepts to approximate and interpret engineering functions.	**L3 (Apply)**
CO4	Analyze multivariable functions using partial derivatives, Jacobians, gradients, and optimization techniques.	**L4 (Analyze)**
CO5	Evaluate areas, volumes, and coordinate transformations using multiple integrals in engineering contexts.	**L5 (Evaluate)**

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	L3	3	3	2	2	1	-	-	-	1	2	-
CO2	L4	3	3	3	2	1	-	-	-	1	2	-
CO3	L3	3	2	2	2	1	-	-	-	-	2	-
CO4	L4	3	3	3	3	2	-	-	-	1	2	-
CO5	L5	3	3	2	3	3	-	1	1	2	2	1

CO-PO Weightage

Average weightage based on the COs is shown below; it is aligned to NBA UG POs PO1 to PO11 and kept within the 1-3 correlation scale.

Avg.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
Avg.	3.0	2.8	2.4	2.4	1.6	0	0.2	0.2	1.0	2.0	0.2

Textbooks

1. B. S. Grewal, *Higher Engineering Mathematics*, 44th Edition, Khanna Publishers, 2017.
2. Erwin Kreyszig, *Advanced Engineering Mathematics*, 10th Edition, Wiley, 2018.
3. G. B. Thomas, M. D. Weir, J. Hass, *Thomas' Calculus*, 14th Edition, Pearson, 2018.
4. R. K. Jain and S. R. K. Iyengar, *Advanced Engineering Mathematics*, 5th Edition, Alpha Science International, 2021.

Reference Books

1. Gilbert Strang, *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2016.
2. David C. Lay, Steven R. Lay, Judi J. McDonald, *Linear Algebra and Its Applications*, 5th Edition, Pearson, 2015.
3. George B. Arfken, Hans J. Weber, Frank E. Harris, *Mathematical Methods for Physicists*, 7th Edition, Academic Press, 2012.
4. Dennis G. Zill and Warren S. Wright, *Advanced Engineering Mathematics*, 6th Edition, Jones & Bartlett, 2018.
5. E. Kreyszig, *Elementary Linear Algebra*, 2nd Edition, Wiley, 1983.

Web Resources & NPTEL Links

1. NPTEL: Linear Algebra – https://nptel.ac.in
2. MIT Open Courseware – Linear Algebra: https://ocw.mit.edu
3. MIT Open Courseware – Multivariable Calculus: https://ocw.mit.edu
4. Python SymPy Documentation: https://docs.sympy.org
5. Google Colab for Python: https://colab.research.google.com
6. SciPy Documentation: https://docs.scipy.org
7. NumPy Documentation: https://numpy.org/doc

Recommended Free & Open-Source AI/Computational Tools

1. Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for matrix computation, visualization, and numerical analysis.
2. SymPy (free, open-source) – For symbolic algebra, calculus, matrix operations, and verification.
3. scikit-learn (free, open-source) – For regression, classification, and exploratory predictive modeling.
4. Jupyter Notebook (free, open-source) – For interactive computation and lab documentation.

5. OpenCV (free, open-source) – For image-based experimentation and visualization support where needed.
6. GNU Octave (free, open-source) – MATLAB-like numerical computing for matrix and calculus workflows.
7. TensorFlow / Keras (free, open-source) – For optional AI demonstrations in regression and classification tasks.

RECOMMENDED FREE & OPEN-SOURCE AI/COMPUTATIONAL TOOLS

1. Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for matrix computation, visualization, and numerical analysis.
2. SymPy (free, open-source) – For symbolic algebra, calculus, matrix operations, and verification.
3. scikit-learn (free, open-source) – For regression, classification, and exploratory predictive modeling.
4. Jupyter Notebook (free, open-source) – For interactive computation and lab documentation.
5. OpenCV (free, open-source) – For image-based experimentation and visualization support where needed.
6. GNU Octave (free, open-source) – MATLAB-like numerical computing for matrix and calculus workflows.
7. TensorFlow / Keras (free, open-source) – For optional AI demonstrations in regression and classification tasks.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
2	0	0	0	2

(26HS0821) COMMUNICATIVE ENGLISH
(Common to all branches)

Course Objectives :

The objectives of the course is to

The objectives of this course are

1. To recall and list fundamental concepts of English grammar, vocabulary, phonetics, and communication models essential for professional contexts.
2. To explain and interpret various forms of written and oral communication including emails, reports, presentations, and group discussions.
3. To use appropriate AI tools and digital platforms to enhance communicative accuracy, fluency, and professional writing skills in real-world scenarios.
4. To examine and differentiate effective versus ineffective communication strategies in professional, academic, and intercultural contexts.
5. To appraise and critique written and spoken communication samples using AI-assisted feedback mechanisms and peer-review methods.
6. To design and produce professional-grade communication artefacts - including portfolios, reports, and presentations - integrating AI tools for quality enhancement.

Unit – I	**Foundations of English Communication**

Content	1. Basics of Communication: Process, Types (Verbal/Non-verbal), and Barriers 2. Parts of Speech: Nouns, Pronouns, Verbs, Adjectives, Adverbs 3. Tenses: Simple, Continuous, Perfect, and Perfect Continuous 4. Subject-Verb Agreement and Correction of Sentences
AI Integration	1. ChatGPT (OpenAI- Grammar explanation, error identification, and instant Q&A on language rules) 2. Grammarly- Real-time grammar, spelling, and punctuation correction with detailed explanations 3. Google Speech-to-Text- Phonetics practice – voice recording for pronunciation and intonation feedback 4. Merriam-Webster Vocabulary Builder- Word-of-the-day, etymology, and interactive vocabulary quizzes

Unit – II	**Listening and Speaking Skills**

Content	1. Principles of Effective Speaking and Active Listening 2. Group Discussion (GD): Techniques, Etiquette, and Evaluation Criteria 3. Presentation Skills: Design Principles, Slide Preparation, and Delivery Techniques

	4. Debate: Argument Construction, Evidence Use, and Rebuttal Techniques 5. Three Questions By Leo Tolstoy
AI Integration	1. Gamma.app / Beautiful.ai- AI-powered presentation design – structured, visual, professional slides 2. Orai (AI Speech Coach)- Fluency analysis, filler-word detection, pacing, and confidence scoring 3. Otter.ai- Auto-transcription and keyword analysis of spoken communication 4. Canva AI- Visual communication design – infographics, posters, and slide aesthetics

Unit – III	**Reading Comprehension & Writing Skills**

Content	1. Reading Strategies: Skimming, Scanning, Intensive and Extensive Reading 2. Comprehension Passages, Synonyms, Antonyms, Words Often Confused, Homonyms, Homophones and Homographs 3. Paragraph Writing: Topic Sentence, Supporting Details, Clincher Sentence 4. Creative Writing – Features and Techniques – Movie Reviews & Blogs
AI Integration	1. QuillBot- AI paraphrasing tool – rewrite passages at varying complexity levels 2. Hemingway Editor- Readability scoring, passive voice detection, and sentence simplification 3. ChatGPT- Essay outline generation, argument structuring, and content brainstorming 4. Rewordify- Vocabulary simplification for improving comprehension of complex texts

Unit – IV	Professional & Business communication

Content	1. Email Writing and Letter Writing: Formal & informal 2. Short Report Writing: Structure, Types (Formal/Informal), and Format 3. Note-Making and Summarization Techniques 4. Resume / CV Writing and LinkedIn Profile Optimization 5. Lazy Portado from Here, There, and Everywhere by Sudha Murthy
AI Integration	1. ChatGPT- Draft and refine professional emails, cover letters, reports, and proposals 2. Kickresume / Resume.io- AI-powered resume builder with ATS-optimized templates 3. Notion AI- Report structuring, agenda writing, and knowledge management 4. Grammarly Business- Tone detection, formality check, and professional register evaluation

Unit – V	Use of AI in Language Learning

Content	1. Artificial Intelligence in Communication, Meaning and basics of AI & Benefits and limitations of AI in human interaction 2.AI chatbots and virtual assistants in education: Use of AI in grammar correction and vocabulary building, AI-based speaking and pronunciation tools, writing assistants, E-Portfolio Creation, Students can also use general LLMs (like ChatGPT or Gemini) to summarize lengthy articles or generate texts on specific topics for reading comprehension. 3.AI in Global Communication: The use of automated translation, summarizing large texts, and finding information in different languages (Suggested Tools: Tools like Google Translate and Reverso are used for real-time and document translation) 4.Digital Communication and AI Ethics: Responsible use of AI in education, Plagiarism and academic honesty, Fake information and AI-generated content, Privacy, bias, and ethical communication online
AI Integration	1. Copyleaks / Turnitin- Plagiarism detection, paraphrase checker, and academic integrity analysis 2. DeepL Translator- High-accuracy translation supporting intercultural communication understanding 3. LinkedIn Learning / Coursera- AI literacy modules – understanding AI bias, ethics, and responsible usage 4. Wix / Adobe Portfolio- E-portfolio creation platform for showcasing communication artefacts.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Students will be able to identify grammatical structures, phonetic patterns, and professional vocabulary, and recall communication frameworks using AI-assisted learning tools.	Remember
CO2	Students will be able to comprehend and interpret written and spoken communication models, and summarize content using AI paraphrasing and summarization tools.	Understand
CO3	Students will be able to compose professional emails, reports, and presentations using AI writing assistants such as ChatGPT, Grammarly, and Hemingway Editor.	Apply
CO4	Students will be able to analyze discourse structures, identify communication barriers, and use AI-driven sentiment analysis tools to evaluate message effectiveness.	Analyze
CO5	Students will be able to evaluate communication samples using AI grammar checkers, readability scorers, and peer-evaluation rubrics aligned to professional standards.	Evaluate
CO6	Students will be able to design and deliver professional communication artefacts - presentations, e-portfolios, and research reports - integrating AI tools for structure, style, and refinement.	Create

CO – PO Articulation Matrix

Scale: 3 – String, 2 – Moderate, 1 – Slight

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	-	-	1	-	1	-	1	1	3	1	2
CO2	1	-	-	1	1	2	1	1	2	3	1	2
CO3	-	-	1	-	3	2	1	2	3	3	2	2
CO4	-	1	1	2	2	2	1	2	2	3	2	2
CO5	-	1	1	2	3	2	2	3	2	3	2	3
CO6	-	1	2	2	3	3	2	3	3	3	3	3
Avg	1.0	1.0	1.3	1.6	2.4	2.0	1.4	2.0	2.2	3.0	1.8	2.3

Text Books

1. Meenakshi Raman & Sangeetha Sharma. Technical Communication: Principles and Practice. 4th Edition, Oxford University Press, 2022.
2. Jyoti Shetty & Rajesh Pareek. Communicative English for Engineers and Professionals. 3rd Edition, Wiley India, 2021.

References Books

1. Rizvi, M. Ashraf Effective Technical Communication. Tata McGraw-Hill,2nd Ed, 2018.
2. Murphy, Herta A. Effective Business Communication. McGraw-Hill Education,7th Ed., 2019.
3. Guffey, Mary Ellen & Dana Loewy. Essentials of Business Communication. Cengage Learning,11th Ed., 2021.
4. Sethi, Anjanee & Bhavana Adhikari, Business Communication.Tata McGraw-Hill, 3rd Ed., 2020.
5. Wren, P.C. & Martin, H. High School English Grammar and Composition. S. Chand Revised Ed., 2019.
6. Tickoo, M.L. & Suresh Kumar, E.R. Writing with a Purpose. Oxford University Press 6th Ed., 2020.

Web References

1. Organization/Author], [Title or Description of Resource], [Website/URL]
2. Organization/Author], [Title or Description of Resource], [Website/URL]
3. https://www.plough.com/articles/the-three-questions
4. https://pubhtml5.com/bbar/lakl/basic/

Other Resource (AI Tool, Open Source Software etc.)

I. Online Learning Resources

1. British Council – Learn English. https://learnenglish.britishcouncil.org Free grammar, vocabulary, listening, and speaking practice modules.
2. Coursera – Communication Skills. https://www.coursera.org/courses?query=communication. University-certified courses on professional and intercultural communication.
3. edX – Writing and Communication https://www.edx.org/learn/writing Courses from MIT and Harvard on academic and business writing.
4. Grammarly Blog https://www.grammarly.com/blog Grammar tips, writing guides, and professional communication resources
5. NPTEL – Communication Skills https://nptel.ac.in/courses/109104101IIT/IISc video lectures on English communication for engineering students.

6. BBC Learning English. https://www.bbc.co.uk/learningenglish Audio, video, and text resources for grammar, vocabulary, and pronunciation
7. Purdue OWL (Online Writing Lab). https://owl.purdue.edu APA/MLA citation guides, grammar rules, and professional writing resources

II. Recommended You Tube Resources

1. TED-Ed: How to Use a Comma. https://youtu.be/GHnl1O3NGwk Grammar – Punctuation.
2. English Speaking Course (Spoken English Guru) https://youtu.be/1OECRXfM8RU Oral Communication.
3. How to Write a Professional Email https://youtu.be/YcqhP_Cv5og. Business Communication.
4. Public Speaking Tips – TED Talk https://youtu.be/iIBMbCFBVN8. Presentation Skills.
5. NPTEL Communication Skills – IIT Kharagpur. https://youtu.be/1Ck4k6rETBo. Academic Communication.
6. How to Write a Resume (Step-by-Step). https://youtu.be/Tt08KmFfIYQ. Professional Writing.
7. Intercultural Communication Explained. https://youtu.be/YMyofREc5Jk. Intercultural Communication.
8. Introduction to AI Writing Tools. https://youtu.be/p_Wq5WNzJcc. AI Literacy.
9. Mind & Mastery. https://youtube.com/@profrjosephponniah?si=u3RBBwVLdf1KeNKW. AI Protocol

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0501) COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON
(Common to all branches)

Prerequisite: None

Course Objectives:

The objectives of the course is to

1. *Introduce students to computational thinking and problem-solving techniques.*
2. *Develop the ability to design algorithms, flowcharts, and pseudocode.*
3. *Provide foundational knowledge of Python programming data types, operators, and I/O.*
4. *Enable students to implement logic using control structures and data structures.*
5. *Develop skills in modular programming, file handling, and basic OOP concepts.*

UNIT – I	**Computational Thinking And Programming Basics**
Content	Computational thinking: characteristics, problem-solving strategies, steps in problem solving; algorithms definition and properties; flowcharts symbols and construction; pseudocode writing and conversion; abstraction, decomposition, pattern recognition, algorithm efficiency basics. Introduction to Python: installation and execution environment; variables, identifiers, keywords; data types, type conversion; input and output statements; expressions and operators arithmetic, relational, logical, assignment; operator precedence. **Source:** TB1: Chapters 1–3 (Spronck computational thinking, algorithms, flowcharts, Python basics); TB2: Chapters 1–2 (Thareja introduction to programming, Python fundamentals)
AI Integration	Use Flowgorithm to design and simulate algorithms (e.g., largest of 3 numbers); draw flowcharts using Lucidchart; use Algorithm Visualizer to trace algorithm execution; use ChatGPT to convert pseudocode to Python and verify output.

UNIT – II	**Decision Making And Looping**
Content	Decision control statements: Boolean expressions; if, if-else, if-elif-else, nested if; conditional expressions (ternary operator). Looping statements: while loop, for loop, iteration techniques, nested loops, infinite loops; loop control break, continue, pass; else with loops. Practical problem solving: prime number check, pattern programs using nested loops, menu-driven programs using control statements.

	Source: TB2: Chapters 4–5 (Thareja decision making, loops, control statements); TB1: Chapters 4–5 (Spronck conditions, loops, iteration)
AI Integration	Use Python Tutor to visualize if-else and loop execution step by step; use Thonny to debug loop-based programs; use Replit for online coding practice; use ChatGPT to generate test cases for control flow programs.

UNIT – III	**Strings And Data Structures**
Content	Strings: representation, indexing, slicing, operations, built-in functions and methods.Lists: creation, indexing and slicing, operations, functions, methods, nested lists.Tuples: creation, operations, packing and unpacking. Sets: creation, set operations union, intersection, difference; frozen sets. **Source:** TB2: Chapters 6–8 (Thareja - strings, lists, tuples, sets, dictionaries); TB1: Chapters 6–7 (Spronck - strings, lists, tuples)
AI Integration	Use NLTK and TextBlob to perform text analysis (sentiment, word frequency) on strings; use Pandas and NumPy for list and array operations; use ChatGPT to explain string slicing and list comprehensions with concrete examples.

UNIT – IV	**Functions And Problem Solving**
Content	Dictionaries: creation, operations, methods; dictionary-based applications. Functions: built-in and user-defined functions; function definition and calling; arguments positional, keyword, default, variable-length; scope of variables (local and global). Recursion: recursive functions factorial, Fibonacci; lambda functions (anonymous functions); applications of functions in problem solving. **Source:** TB2: Chapters 9–10 (Thareja functions, recursion, lambda, dictionaries); TB1: Chapters 8–9 (Spronck functions, recursion, problem solving)
AI Integration	Use Jupyter Notebook for interactive function experiments; use SymPy for symbolic mathematical expressions; use PyCharm for modular program development; use ChatGPT to trace recursive function calls step by step.

UNIT – V	**File Handling, Exception Handling, And Oop**
Content	Modules and packages: creating and importing modules; standard library modules. File handling: opening, reading, writing, and closing files; file modes; working with text files; processing CSV and Excel files. Exception handling: types of errors (syntax, runtime, logical); try, except,

	finally blocks; raising exceptions. Introduction to OOP: classes, objects, attributes, methods, constructors, self keyword; basic applications of OOP in Python. **Source:** TB2: Chapters 11–13 (Thareja modules, file handling, exception handling, OOP); TB1: Chapter 10 (Spronck file handling, exceptions, object-oriented programming)
AI Integration	Use Pandas and OpenPyXL to read and write CSV/Excel files; use Pytest for testing exception handling; use ChatGPT to explain OOP class design with real-world examples; use Django for a simple mini web application as a capstone project.

Course Outcomes (Cos):

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Apply computational thinking concepts to design algorithms, flowcharts, and basic Python programs.	L3 (Apply)
CO2	Develop programs using decision-making and looping constructs to solve logical problems.	L3 (Apply)
CO3	Analyse and manipulate data using strings, lists, tuples, and sets for problem solving.	L4 (Analyse)
CO4	Design modular programs using functions, recursion, and dictionaries to improve code reusability.	L4 (Analyse)
CO5	Implement file handling, exception handling, and basic object-oriented programming concepts in Python applications.	L3 (Apply)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define algorithm, pseudocode, flowchart; describe data types and operators.
L3	Apply	**CO1–CO5**	Write Python programs using control structures, functions, and file handling.
L4	Analyse	**CO3, CO4**	Compare list vs. tuple vs. set; analyse recursive vs. iterative solutions.
L5	Evaluate	**CO5**	Evaluate exception handling strategies for a given fault scenario.
L6	Create	**CO4, CO5**	Design a modular OOP-based Python application for a real-world problem.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO 1	L3	**3**	**2**	1	1	1	1	–	–	–	1	–
CO 2	L3	**3**	**3**	**2**	1	1	–	–	–	–	1	–
CO 3	L4	**3**	**3**	**2**	**2**	**2**	–	–	–	–	1	–
CO 4	L4	**3**	**3**	**3**	**2**	**2**	–	–	–	1	**2**	–
CO 5	L3	**3**	**3**	**2**	1	**2**	–	–	–	–	1	–

CO-PO Correlation Highlights	Strongest correlation with PO1 (Engineering Knowledge) through Python programming fundamentals and PO2 (Problem Analysis) through algorithm design. PO3 (Design/Development) is addressed in CO4 and CO5 through modular and OOP-based program design. PO5 (Modern Tools) through Python IDE, Flowgorithm, and Python Tutor across all UNITs.

Textbooks :

1. Pieter Spronck, *Computational Thinking with Python*, 2nd Edition, Springer, 2021.
2. Reema Thareja, *Programming in Python*, Oxford University Press, 2017.

Reference Books :

1. Allen B. Downey, *Think Python: How to Think Like a Computer Scientist*, 2nd Edition, O'Reilly Media, 2015.
2. Eric Matthes, *Python Crash Course*, 2nd Edition, No Starch Press, 2019

Online Resources :

- Algorithm & Flowchart: Flowgorithm , flowgorithm.org,
- Lucidchart , lucidchart.com ,
- Algorithm Visualizer , algorithm-visualizer.org
- Programming & Debugging: Python Tutor – pythontutor.com
- Thonny – thonny.org
- Replit – replit.com
- Data & Scientific: Pandas – pandas.pydata.org
- NumPy – numpy.org
- SymPy – sympy.org

- NLP & Text: NLTK – nltk.org
- TextBlob – textblob.readthedocs.io
- Development: Jupyter Notebook – jupyter.org
- PyCharm – jetbrains.com/pycharm
- OpenPyXL – openpyxl.readthedocs.io
- NPTEL – Problem Solving through Programming in C /
- Python – nptel.ac.in

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
1	0	4	0	3

(26CS3101) AI TOOLS AND APPLICATIONS
(Common to all branches)

Prerequisite: None

Course Objectives;

The objectives of the course is to

1. *Introduce students to AI concepts and the landscape of AI tools available for everyday academic and professional use.*
2. *Develop practical skills in using AI for document creation, presentations, writing, and academic tasks applicable across all engineering disciplines.*
3. *Build proficiency in AI-powered data analysis, spreadsheet automation, and intelligent research tools without requiring programming knowledge.*
4. *Expose students to AI tools in visual design, media creation, and note-taking, and progressively to AI applications specific to their engineering branch.*
5. *Cultivate responsible AI habits: verifying outputs, protecting privacy, respecting intellectual property, and understanding ethical and legal boundaries of AI use.*

UNIT – I	**Understanding Ai And Ai For Documents And Presentations**
Content	What is AI: AI vs. ML vs. Deep Learning vs. Generative AI; brief history (Turing Test → ChatGPT → generative AI era); how LLMs work (training data, next-token prediction, temperature, hallucinations, context windows, knowledge cutoffs). AI ecosystem: ChatGPT, Claude, Gemini, Microsoft Copilot strengths and differences. AI for presentations: Gamma.app full slide deck from a paragraph prompt; Microsoft Copilot in PowerPoint slide generation, design suggestions, speaker notes; Canva AI Magic Design, text-to-presentation, brand kits; Beautiful.ai smart slide layouts; comparing AI-generated vs. manually created presentations. AI for document writing: Microsoft Copilot in Word draft, rewrite, summarize; Gemini in Google Docs writing assistance and inline Q&A; Notion AI meeting notes, project plans, knowledge base; Claude Projects organizing documents and having a persistent AI assistant over a set of files. AI writing assistance: Grammarly grammar, tone, clarity, plagiarism; ChatGPT and Claude for lab reports, proposals, and letters; academic integrity AI as an assistant not an author; disclosure norms; identifying AI-generated content using GPTZero.

AI Integration	Use Gamma.app to generate a presentation on an engineering topic in under 2 minutes; use Copilot in Word or Gemini in Docs to draft and improve a report; compare ChatGPT, Claude, Gemini, and Microsoft Copilot on the same writing task; use Grammarly to review your own writing.

UNIT – II	**Ai For Research, Note-Taking, Data, And Spreadsheets**
Content	NotebookLM (Google): uploading lecture notes, textbooks, PDFs, and research papers as sources; asking questions grounded in uploaded documents; generating study guides, timelines, briefing documents, and audio overviews (podcast-style summaries) from sources; source-grounded answers that cite the exact passage eliminating hallucination risk. AI for research: Perplexity AI (real-time web search with citations), Gemini Deep Research (multi-step research reports), Elicit and Consensus (academic paper search with evidence summaries); verifying AI-generated citations on Google Scholar; why AI fabricates references and how to detect it. Prompt engineering: zero-shot, one-shot, few-shot, chain-of-thought, role-based prompting; iterative refinement; negative prompting; building effective prompts for summarization, comparison, explanation, and data extraction; using ChatGPT custom instructions and Claude Projects for persistent context. AI for spreadsheets and data analysis: Microsoft Copilot in Excel VLOOKUP, IF, COUNTIF, pivot tables from natural language; Gemini in Google Sheets formula generation and chart creation; ChatGPT Advanced Data Analysis and Julius AI uploading CSV files, computing statistics, generating charts, interpreting patterns no coding required.
AI Integration	Upload your own lecture notes to NotebookLM and generate a study guide and audio overview; use Perplexity AI and Gemini Deep Research to research a topic; verify citations on Google Scholar; use Wolfram Alpha or Symbolab to solve and verify equations from your current semester subjects; use Copilot in Excel or ChatGPT ADA to analyse a real dataset.

UNIT – III	**Ai For Visual Design, Diagrams, And Media**
Content	AI for image generation: DALL-E 3 (via ChatGPT), Canva AI (text-to-image, Magic Design, background remover), Adobe Firefly, Ideogram writing effective image prompts (style, subject, composition, colour); iterative prompt refinement for posters, diagrams, and mockups. AI for diagrams and visual thinking: Napkin.ai converting text or concepts into visual diagrams, charts, and infographics automatically; Whimsical AI AI-generated mind maps, flowcharts, and wireframes from a description; Miro AI collaborative whiteboard with AI summarization and diagramming.

	AI for audio and video: ElevenLabs text-to-speech and voice cloning for presentations and voiceovers; Suno and Udio AI music generation for project videos; Runway and Pika generating short video clips from text prompts; HeyGen AI avatars for video presentations; current limitations and appropriate use cases. AI content detection and critical evaluation: GPTZero, Originality.ai for AI-text detection; Google Lens and reverse image search for AI-generated images; fact-checking workflows cross-referencing primary sources; deepfakes and synthetic media recognition, risks, and responsible creation.
AI Integration	Use Napkin.ai to convert a paragraph of text into a diagram; use Whimsical AI to generate a mind map of a course topic; use DALL-E 3 or Canva AI to create an event poster through 3 prompt iterations; use ElevenLabs to create a voiceover; test GPTZero on 5 samples and analyse confidence scores.

UNIT – IV	**Ai Across Engineering Disciplines**
Content	AI in Mechanical Engineering: generative design in CAD (Autodesk Fusion 360 AI-optimized structures for weight and load); predictive maintenance using sensor data and ML; AI-assisted FEA simulation; manufacturing process optimization; digital twins. AI in Civil and Electrical Engineering: structural health monitoring using ML; smart grid load forecasting and energy management; AI for circuit simulation and PCB layout optimization (Altium AI); VLSI design automation floor planning and routing; power systems fault detection. AI in Electronics and Communication: AI-assisted signal processing and filtering; speech recognition for embedded systems; computer vision for industrial quality inspection; AI in antenna design and spectrum management; MATLAB AI and Deep Learning Toolbox. AI in Computer Science and allied branches: GitHub Copilot for code generation, explanation, and review; AI-powered testing (Testim, Selenium AI); AI in cybersecurity anomaly detection and intrusion detection; no-code/low-code platforms (Bubble, Glide, Webflow AI); AI for data science (Jupyter AI, DataRobot).
AI Integration	Each student identifies 5 real AI tools in their own engineering branch using Claude or ChatGPT and verifies each using official product pages or research publications; present findings in a 3-minute class presentation with a structured comparison table.

UNIT – V	**Responsible Ai, Ethics, And Capstone Project**
Content	AI bias and fairness: sources of bias (data, model, deployment); real

	documented cases Amazon hiring algorithm (2018), COMPAS recidivism (2016), facial recognition disparities (NIST 2019); fairness metrics; misinformation, deepfakes, and synthetic media risks. Privacy and intellectual property: what happens to data entered into public AI tools; PII risks; data retention policies of ChatGPT (OpenAI), Claude (Anthropic), and Gemini (Google); who owns AI-generated content; copyright status globally; academic plagiarism with AI; personal data safety protocol. AI governance frameworks: EU AI Act risk tiers (unacceptable, high, limited, minimal); UNESCO AI Ethics Recommendation; India's NITI Aayog Responsible AI Guidelines; India Digital Personal Data Protection Act 2023 (DPDP Act); environmental cost of AI energy and carbon footprint of LLM training and inference. Capstone Mini-Project: each student (or pair) defines a real engineering problem, selects at least 3 AI tool categories from UNITs I–IV, produces a quality-verified deliverable (report, presentation, diagram, or automation), and presents in 5 minutes. Evaluation: problem clarity, tool selection, output quality, human refinement, responsible AI disclosure.
AI Integration	Read and compare data retention policies of ChatGPT, Claude, and Gemini; classify 5 daily-use AI systems using EU AI Act risk tiers; use GPTZero and Google Lens to detect AI-generated content; design and present the Capstone Mini-Project integrating tools from all five UNITs.

Course Outcomes (Cos):

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Explain what AI is, how it works conceptually, and describe the landscape of AI tools across text, image, audio, video, and code categories. (L2)	L3 (Apply)
CO2	Use AI tools to create documents and presentations; apply prompt engineering and NotebookLM to improve writing, research, and study habits. (L3)	L4 (Analyse)
CO3	Perform data analysis using AI without coding; use AI-powered research and note-taking tools to find, verify, organize, and summarize information. (L3)	L4 (Analyse)
CO4	Apply AI tools in visual design, diagrams, and media; identify and verify AI applications relevant to their own engineering discipline. (L4)	L5 (Evaluate)
CO5	Demonstrate responsible AI usage bias, privacy, ethics, governance; design and present an AI-assisted mini-project using tools from all UNITs. (L5)	L5 (Evaluate)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define AI, GenAI, LLM; describe prompt types and tool categories.
L3	Apply	**CO1–CO3**	Use AI tools to create presentations, research summaries, and visual diagrams.
L4	Analyse	**CO2, CO3**	Evaluate prompt effectiveness; compare tools for a given task.
L5	Evaluate	**CO4, CO5**	Assess tool appropriateness per discipline; evaluate ethical implications.
L6	Create	**CO1, CO5**	Design an AI-assisted project deliverable with responsible usage documentation.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO1	L3	**3**	**2**	**2**	1	**3**	**2**	–	1	1	**2**	1
CO2	L4	**3**	**3**	**2**	**2**	**3**	1	–	1	1	**2**	1
CO3	L4	**3**	**2**	**3**	**2**	**3**	1	–	1	1	**2**	1
CO4	L5	**3**	**3**	**3**	**3**	**3**	**2**	–	1	**2**	**2**	**2**
CO5	L5	**3**	**2**	**2**	**2**	**3**	**2**	1	**2**	**2**	**2**	**2**

CO-PO Correlation Highlights	Strong PO5 (Modern Tools) through Gamma.app, NotebookLM, ChatGPT, Claude, Napkin.ai, and domain-specific AI tools across all UNITs. PO3 (Design/Development) through CO3 (visual design and diagram creation). PO6 (Engineer & Society) and PO8 (Ethics) through responsible AI usage, bias awareness, and the capstone ethics module in CO5.

Reference Books:

1. Ethan Mollick, *Co-Intelligence: Living and Working with AI*, Portfolio/Penguin, 2024.
2. Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson Education, 2020.
3. Reid Blackman, *Ethical Machines*, Harvard Business Review Press, 2022.

Online Resources:

1. https://www.youtube.com/@JoyofLearningCSE/featured
2. EU AI Act – eur-lex.europa.eu | UNESCO AI Ethics – unesco.org | NITI Aayog – niti.gov.in
3. India DPDP Act 2023 – meity.gov.in | ProPublica COMPAS – propublica.org | MIT Technology Review – technologyreview.com |

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0502) COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON LAB
(Common to all branches)

Prerequisite: None

Course Objectives

The objectives of the course is to

1. *Introduce students to computational thinking and problem-solving techniques.*
2. *Develop the ability to design algorithms, flowcharts, and pseudocode.*
3. *Provide foundational knowledge of Python programming including data types, operators, and input/output.*
4. *Enable students to implement logic using control structures and Python data structures.*
5. *Develop skills in modular programming, file handling, and basic object-oriented programming.*

List of Experiments

MODULE I: Computational Thinking & Programming Basics

Topics: Computational thinking decomposition and abstraction; algorithms, flowcharts, and pseudocode; Python environment setup; variables, data types, input/output, and operators; basic computational problems.

AI Integration: Generate alternate solutions and test inputs; verify flowchart correctness; compare solution efficiency; explain type conversion.

Suggested Tools: Python, Flowgorithm, draw.io AI, ChatGPT, Microsoft Copilot

Experiments:

1. **Experiment 1: Problem Solving using Computational Thinking** Select three daily-life problems; apply decomposition and abstraction; write algorithms, draw flowcharts, and convert them into pseudocode.
2. **Experiment 2: Flowchart Design and Algorithm Simulation** Design flowcharts for largest of three numbers, student grade calculation, and electricity-bill generation; simulate execution and verify outputs.
3. **Experiment 3: Python Programming Fundamentals** Install the Python environment; write programs using variables, data types, input/output, and operators; solve simple interest, area, and UNIT-conversion problems.

MODULE II: Decision Making & Looping

Topics: Decision-control statements; looping and iterative problem solving; nested loops and pattern generation.

AI Integration: Generate boundary and test cases; verify decision logic; trace loop execution; compare loop alternatives; optimize nested loops.

Suggested Tools: Python, Python Tutor, ChatGPT, Microsoft Copilot

Experiments:

1. **Experiment 4: Decision Control Statements** Develop Python programs for largest among three numbers, grade calculation, leap-year checking, and a menu-driven calculator.
2. **Experiment 5: Looping and Iterative Problem Solving** Implement programs for prime-number generation, Armstrong numbers, factorial computation, and multiplication tables.
3. **Experiment 6: Nested Loops and Pattern Generation** Generate number, pyramid, diamond, and character patterns.

MODULE III: Strings & Data Structures

Topics: String processing; lists and nested lists; tuples and sets.

AI Integration: Generate string test cases and sample datasets; explain slicing and immutable structures; compare list approaches and suggest optimizations.

Suggested Tools: Python, Python Tutor, Algorithm Visualizer, ChatGPT

Experiments:

1. **Experiment 7: String Processing Applications** Develop programs for string reversal, palindrome checking, frequency counting, and word-occurrence analysis.
2. **Experiment 8: List and Nested List Operations** Perform sorting, searching, list merging, and matrix operations using nested lists.
3. **Experiment 9: Tuple and Set Applications** Implement tuple packing/unpacking, student-record storage, and set union, intersection, and difference operations.

MODULE IV: Functions & Problem Solving

Topics: User-defined functions; recursion; dictionaries and lambda functions.

AI Integration: Generate function examples and datasets; verify outputs; generate recursion trees; compare iterative and recursive solutions; explain lambda execution.

Suggested Tools: Python, Python Tutor, Algorithm Visualizer, ChatGPT

Experiments:

1. **Experiment 10: User-Defined Functions** Implement positional, keyword, default, and variable-length arguments.
2. **Experiment 11: Recursive Problem Solving** Develop recursive solutions for factorial, Fibonacci, GCD, and Tower of Hanoi.

3. **Experiment 12: Dictionary and Lambda Applications** Implement a student database, dictionary operations, dictionary sorting, and lambda-based computations.

MODULE V: File Handling, Exception Handling & Oop

Topics: File processing and data storage; exception handling and testing; object-oriented programming classes, objects, constructors, methods.

AI Integration: Generate datasets and error scenarios; verify file outputs; recommend exception strategies; generate class diagrams and suggest object interactions.

Suggested Tools: Python, ChatGPT, Microsoft Copilot

Experiments:

1. **Experiment 13: File Processing and Data Storage** Develop programs to create and read text files, count words, and process CSV data.
2. **Experiment 14: Exception Handling and Testing** Implement try-except, finally, and custom exceptions; apply exception handling to division operations, file access, and input validation.
3. **Experiment 15: Object-Oriented Programming Mini Project** Develop a Python application using classes, objects, constructors, and methods.

Suggested Mini Projects:

1. Student Management System
2. Library Management System
3. Banking System

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply computational thinking concepts to design algorithms, flowcharts, and Python programs.	L3 (Apply)
CO2	Develop programs using decision-making and looping constructs to solve logical problems.	L3 (Apply)
CO3	Analyse and manipulate data using strings, lists, tuples, and sets.	L4 (Analyze)
CO4	Design modular programs using functions, recursion, and dictionaries.	L4 (Analyze)
CO5	Implement file handling, exception handling, and object-oriented concepts in Python applications.	L3 (Apply)

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO 9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning
CO 1	L3	3	3	2	1	3	–	–	–	–	1	–	2
CO 2	L3	3	3	2	1	3	–	–	–	–	1	–	2
CO 3	L4	3	3	2	2	3	–	–	–	–	1	1	2
CO 4	L4	3	3	3	2	3	–	–	–	1	1	1	2
CO 5	L3	3	2	3	2	3	–	–	–	1	1	1	2

Textbooks :

1. Pieter Spronck, *Computational Thinking with Python*.
2. Reema Thareja, *Programming in Python*, Oxford University Press.

Reference Books :

1. Allen B. Downey, *Think Python: How to Think Like a Computer Scientist*.
2. Eric Matthes, *Python Crash Course*.
3. John Zelle, *Python Programming: An Introduction to Computer Science*.

Recommended AI / Computational Tools :

1. AI Assistants: ChatGPT, Microsoft Copilot – code generation, explanation and test-case generation.
2. Flowchart & Diagramming: Flowgorithm, Lucidchart AI, draw.io AI – algorithm and flowchart design.
3. Visualization: Algorithm Visualizer, Python Tutor – execution tracing and recursion visualization.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26HS0850) PHYSICS FOR ADVANCED COMPUTING LAB

Prerequisites: Engineering Physics (Theory)

Course Objectives

1. Determine optical properties of materials (dispersive power, refractive index, wavelength) using prisms, diffraction gratings, and interference techniques.
2. Study wave phenomena in vibrating strings, tuning forks, and coupled oscillators.
3. Investigate elastic and mechanical properties of materials using Hooke's law and torsional pendulum methods.
4. Apply quantum concepts by estimating Planck's constant through the photoelectric effect.
5. Employ laser-based and ultrasonic techniques to measure particle size and acoustic velocity in liquids.

List of Experiments

1. Determination of the dispersive power of the prism.
2. Determination of the thickness of a thin object by the wedge method
3. Determination of the radius of curvature of the lens by Newton’s rings
4. Determination of wavelengths of various colours of mercury spectrum using a diffraction grating in the normal incidence method
5. Determination of wavelength using a diffraction grating
6. Determination of particle size using a diode laser
7. Estimation of Planck’s constant using the photoelectric effect
8. Determination of spring constant by Hooke’s law
9. Determination of the frequency of the tuning fork-Melde’s experiment
10. Verification of the three laws of stretched strings using sonometer
11. Determination of frequency of normal modes of the Coupled Oscillator
12. Study of the Refractive Index of Materials using a hollow prism used in surveying
13. Study of oscillations of a mass under different combinations of springs (Loaded Springs)
14. Determination of ultrasonic velocity in liquid (Acoustic grating)
15. Determination of Rigidity modulus of material of a wire-dynamic method (Torsional pendulum)

Note: Any **TEN** of the listed experiments are to be conducted.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Explain the physical principles underlying the optics, mechanics, waves, oscillations, and quantum physics through laboratory experiments.	L2 (Understand)
CO2	Conduct experiments to determine optical constants such as dispersive power, wavelength and radius of curvature	L3 (Apply)
CO3	Measure oscillations, and quantum physics related properties of materials including particle size, Planck's constant, spring constant, and rigidity modulus.	L3 (Apply)
CO4	Analyse experimental data, estimate errors, and validate results using Python-based computational tools.	L4 (Analyze)
CO5	Evaluate the agreement between experimental and theoretical values using AI/ML-assisted data analysis.	L5 (Evaluate)

CO – PO Articulation Matrix
Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	–	2	1	–	–	–	–	1	–	2
CO2	3	3	1	2	2	–	–	–	1	2	1	2
CO3	3	3	2	3	2	–	1	–	2	2	1	1
CO4	2	3	2	3	2	–	1	1	2	2	2	1
CO5	3	3	2	3	3	1	1	1	3	3	2	1

AI Extension: Python (NumPy, SciPy, OpenCV, Matplotlib) – FREE on Google Colab for optics and EM simulations

Online Learning Resources

1. HyperPhysics – hyperphysics.phy-astr.gsu.edu
2. PhET Simulations – phet.colorado.edu
3. https://iitb.vlabs.co.in/discipline.html?discipline=Physical_Sciences

Recommended AI Tools

Python (NumPy, SciPy, OpenCV, Matplotlib) – FREE on Google Colab for wave optics and signal analysis

TensorFlow / Keras (free, open-source) – For crack and defect detection CNNs on Google Colab

scikit-learn (free, open-source) – For anomaly detection in vibration and NDT data

PhET Interactive Simulations (free) – For wave optics, quantum mechanics and oscillations

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	2	1

(26HS0822) COMMUNICATIVE ENGLISH LAB
(Common to all branches)

Course Objectives

The objectives of this course are

1. To develop students' oral communication skills through structured speaking and listening exercises.
2. To enhance pronunciation, intonation, and fluency using AI-assisted speech tools.
3. To build professional communication competencies including group discussion, debate, and presentation skills.
4. To enable students to write effective professional documents such as emails, reports, and cover letters.
5. To integrate AI tools in language learning for self-assessment, feedback, and continuous improvement

List of Experiments

Part A – Regular Experiment

1. Phonetics & Pronunciation Practice
2. Listening Comprehension &Note-Taking
3. Spoken English & Fluency Building
4. Group Discussion Techniques
5. Public Speaking & Presentations
6. Interview Skills & Mock Interviews
7. Professional Email & Business Writing
8. Resume & Cover Letter Writing
9. Reading Comprehension & Critical Thinking
10. Debate & Argumentation Skills
11. Report Writing & Technical Communication
12. Etiquette, Netiquette & Personal Branding

Part B – AI Integration Lab Exercises

1. Use AI speech-to-text to detect mispronunciations; real-time phoneme feedback via ELSA Speak
2. Auto-generate transcripts; compare student notes with AI-generated summaries using Otter.ai

3. AI analyses speech for filler words, pace, and clarity; ChatGPT prompts for spontaneous speaking
4. AI evaluates argument structure and language; ChatGPT generates counter-arguments for practice
5. AI slide deck generation; real-time delivery feedback on pacing and confidence
6. AI-powered mock interview with instant feedback on content, grammar, and tone
7. AI refines email drafts; tone & clarity check; compare student draft with AI-improved version
8. AI resume scanner; keyword optimisation suggestions; ChatGPT cover letter generation and critique
9. AI generates comprehension questions; text summarisation and comparative analysis
10. AI generates arguments on both sides; evaluate logical structure using ChatGPT
11. AI improves report coherence and passive/active voice usage; grammar and structure feedback
12. AI reviews LinkedIn summaries; digital persona analysis; personal brand statement via ChatGPT

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Pronounce English words correctly using standard phonemic symbols and demonstrate improved speech fluency.	Apply
CO2	Comprehend spoken English through active listening and respond appropriately in formal and informal contexts.	Understand
CO3	Participate effectively in group discussions, presentations, and debates demonstrating logical argumentation.	Analyse
CO4	Write grammatically correct and contextually appropriate professional documents including emails, reports, and resumes.	Create
CO5	Utilize AI-powered tools for language learning, self-evaluation, and continuous improvement of English skills.	Evaluate

CO – PO Articulation Matrix

Scale: 3 – String, 2 – Moderate, 1 – Slight

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	2	-	1	2	-	-	-	-	-
CO2	2	-	-	2	-	2	1	-	-	-	-	-
CO3	2	-	-	3	-	3	2	-	-	-	-	-
CO4	2	-	-	2	-	2	1	-	-	-	-	-
CO5	1	-	-	2	3	2	2	-	-	-	-	-

Text Books

1. Raman, Meenakshi & Sharma, Sangeeta. Technical Communication: Principles and Practice (3rd Ed.). Oxford University Press, 2022.
2. Krishnaswamy, N. & Krishnaswamy, L. The Story of English in India. Foundation Books, Cambridge University Press, 2021.

References Books

1. Swan, Michael. Practical English Usage (4th Ed.). Oxford University Press, 2016.
2. Murphy, Raymond. English Grammar in Use (5th Ed.). Cambridge University Press, 2019.
3. Rizvi, M. Ashraf. Effective Technical Communication (2nd Ed.). McGraw-Hill Education, 2018.
4. Turton, N.D. & Heaton, J.B. Longman Dictionary of Common Errors. Pearson Longman, 2017.
5. Lesikar, Raymond V. & Flatley, Marie E. Basic Business Communication (12th Ed.). McGraw-Hill, 2020.
6. Rutherford, Andrea J. Basic Communication Skills for Technology (3rd Ed.). Pearson, 2019.

Web References
Online Resources

1. BBC Learning English – https://www.bbc.co.uk/learningenglish (Grammar, pronunciation, and listening)
2. British Council Learn English – https://learnenglish.britishcouncil.org
3. TED Talks – https://www.ted.com/talks (Listening & speaking practice)
4. Coursera English Communication Skills Specialisation – https://www.coursera.org
5. edX English for Professional and Academic Purposes – https://www.edx.org
6. Grammarly Blog – https://www.grammarly.com/blog (Writing tips)
7. ELSA Speak (App) – https://elsaspeak.com (AI pronunciation coach)

You Tube Channels

1. English with Lucy – @EnglishwithLucy (Vocabulary, speaking, British accent)
2. Learn English with TV Series – @LearnEnglishWithTVSeries
3. Rachel's English – @rachelsenglish (American accent and pronunciation)
4. TED-Ed – @TEDEd (Critical thinking and communication ideas)
5. Spoken English Hub – @SpokenEnglishHub (Indian context, interview prep)
6. JenniferESL – @JenniferESL (Grammar and business English)

Other Resource (AI Tool, Open Source Software etc.)

1. ELSA Speak- AI Pronunciation Coach App
2. Grammarly- AI Writing Assistant
3. ChatGPT / Claude.ai- Generative AI Chatbot
4. Otter.ai- AI Transcription Tool
5. Yoodli - AI Public Speaking Coach
6. Hemingway Editor- Writing Clarity Tool
7. Mentimeter- Interactive Presentation
8. Audacity- Audio Recording Software
9. Google Meet / Zoom - Video Conferencing
10. Canva / Beautiful.ai- AI Presentation Design

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	1	0.5

(26HS0823) NSS / NCC / Scouts & Guides / Community Service
(Common to all branches)

Course Objectives

The objectives of this course are to

1. Inculcate the spirit of volunteerism, patriotism and social responsibility among students.
2. Enable students to understand the structure, objectives and working of NSS, NCC, Scouts & Guides.
3. Develop leadership, communication, teamwork and crisis-management skills through experiential learning.
4. Equip students with basic first-aid and disaster-preparedness competencies.
5. Promote community engagement and sustainable development values through organised service activities.

Unit-wise Syllabus with Learning Outcomes

UNIT 1	**Foundation of NSS, NCC, Scouts & Guides and Community Service**
Topics	Introduction to NSS – history, objectives, badge, motto, pledge Introduction to NCC – ranks, uniforms, aims, NCC Act 1948 Scouts & Guides – Baden-Powell legacy, Bharat Scouts & Guides Act Community Service – concept, importance and types Citizenship, civic sense and social responsibility Volunteerism: global and Indian perspective (NYKS, NSS, Red Cross) Legal & ethical framework for community service
Unit-wise Learning Outcomes (ULOs)	On completion of this unit, students will be able to: ULO1.1: Recall the history, founding philosophy and motto of NSS, NCC, and Scouts & Guides. [Remember] ULO1.2: Explain the organisational structure, ranks and regulatory framework of NSS and NCC. [Understand] ULO1.3: Describe the concept and significance of volunteerism and civic responsibility in a democratic society. [Understand] ULO1.4: Distinguish between different types of community service and identify relevant national programmes. [Analyse]

UNIT 2	Leadership, Skill Development, Camping and Disaster Management
Topics	Leadership development – theories and models (situational, transformational) Team building, communication & conflict resolution Parade, drill & physical training basics Map reading, navigation and field craft (NCC) Camp organisation – planning, logistics, safety protocols Adventure activities: trekking, rock-climbing, swimming (awareness) Disaster management – concepts, types, roles of NSS/NCC volunteers First Aid – CPR, wound management, fractures, snake bite, burn
Unit-wise Learning Outcomes	On completion of this unit, students will be able to: ULO2.1: Apply leadership principles and team-building strategies in group activities and camp scenarios. [Apply]

(ULOs)	ULO2.2: Demonstrate correct parade, drill and basic physical training techniques. [Apply] ULO2.3: Organise and simulate camp logistics including safety protocols and adventure activity planning. [Apply] ULO2.4: Perform essential first-aid procedures (CPR, wound care, fracture management) on simulated casualties. [Apply] ULO2.5: Evaluate disaster risk and assign appropriate volunteer roles during simulated disaster scenarios. [Evaluate]

UNIT 3	Community Projects, Social Awareness and Service Documentation
Topics	Community needs assessment – participatory methodology Project planning: Swachh Bharat, Pulse Polio, Blood Donation camps Literacy campaigns and awareness drives (Padhna Likhna Abhiyan) Environmental conservation – plantation, solid waste, water conservation Gender sensitisation and women empowerment programmes Digital literacy and e-governance outreach Documentation and reporting of service-learning activities Reflection, portfolio preparation and field report writing
Unit-wise Learning Outcomes (ULOs)	On completion of this unit, students will be able to: ULO3.1: Conduct a community needs assessment using participatory tools. [Analyse] ULO3.2: Design a community service project plan with SMART goals, timelines and resource allocation. [Create] ULO3.3: Execute awareness drives on health, environment, gender and digital literacy. [Apply] ULO3.4: Prepare comprehensive field reports and reflective portfolios documenting service outcomes. [Create] ULO3.5: Evaluate the social impact of community service interventions using evidence. [Evaluate]

Course Outcomes (COs)

CO No.	Course Outcome	Bloom's Level
CO1	Understand the philosophy, history, and objectives of NSS/NCC/Scouts & Guides and community service programmes.	Remember/Understand
CO2	Demonstrate leadership, teamwork, and organisational skills through camp/parade/drill activities.	Apply
CO3	Plan and execute community service projects addressing social issues like health, hygiene, literacy, and environment.	Analyse/Apply
CO4	Apply first aid, disaster-preparedness, and civic responsibilities in real-world scenarios.	Apply
CO5	Develop civic sense, national integration, and sustainable community-development values.	Evaluate

Bloom's Taxonomy Levels: Remember (L1) | Understand (L2) | Apply (L3) | Analyse (L4) | Evaluate (L5) | Create (L6)

Text Books

S.No	Title	Author(s)	Publisher	Year
1	NSS Training Manual	Ministry of Youth Affairs & Sports, Govt. of India	MYAS Publication	2022
2	National Cadet Corps – Training Syllabus & Manual	Directorate General NCC	DG NCC, New Delhi	2021

Reference Books

S.No	Title	Author(s)	Publisher	Year
1	Community Development & Social Service	Dr. S. K. Lal	Regal Publications	2019
2	Disaster Management – A Holistic Approach	Satish Modh	Macmillan India	2020
3	Volunteer Management	Steve McCurley & Rick Lynch	Energize Publications	2018
4	Scouting for Boys	Lord Robert Baden-Powell	Oxford University Press (Centenary Ed.)	2008

Web Resources

1. NSS Official Portal – https://nss.gov.in
2. NCC Official Site – https://indiancc.mygov.in
3. Bharat Scouts & Guides – https://bsgindia.org
4. UN Volunteers – https://www.unv.org
5. NYKS (Nehru Yuva Kendra) – https://nyks.nic.in
6. National Disaster Management Authority – https://ndma.gov.in
7. Swachh Bharat Mission – https://swachhbharat.mygov.in
8. IndiaVolunteers – https://www.indiavolunteers.org

YouTube Resources

1. NSS Song & Pledge (Official MYA&S) – https://youtu.be/nss_official_01
2. NCC Drill & Parade Training Basics – https://youtu.be/ncc_drill_02
3. Community Service Project Planning – https://youtu.be/community_svc_03
4. First Aid & CPR Tutorial – Red Cross India – https://youtu.be/firstaid_rci_04
5. Disaster Management Awareness – NDMA – https://youtu.be/ndma_dm_05
6. Baden-Powell and the Scout Movement – https://youtu.be/scouts_bp_06
7. Leadership Skills for Youth – https://youtu.be/leadership_youth_07
8. Swachh Bharat Campaign Activities – https://youtu.be/swachh_sb_0

CO – PO / PSO Mapping

Correlation: 3 = Strong | 2 = Moderate | 1 = Slight | – = No correlation

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	-	-	-	3	3	2	2	-	-	-	2	-
CO2	-	-	3	-	-	3	2	3	3	-	-	-	2	2
CO3	-	-	2	3	2	3	3	2	3	-	-	-	-	-
CO4	-	-	3	2	-	2	2	3	3	-	-	-	-	-
CO5	-	-	-	-	3	3	3	2	2	-	-	-	-	-

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
2	0	0	0	0

(26CS4801) CYBERSECURITY AWARENESS FOR ENGINEERS

Prerequisite: None

Course Objectives:

The objectives of this course is to

1. *Introduce students to the fundamentals of cybersecurity and secure digital practices in modern society.*
2. *Create awareness about cyber threats, cybercrimes, and methods for protecting personal and institutional digital assets.*
3. *Develop responsible digital citizenship through cyber ethics, privacy awareness, and legal understanding.*
4. *Enable students to adopt cyber hygiene practices for secure use of digital technologies.*
5. *Familiarize students with emerging cybersecurity concerns in cloud, digital platforms, and AI-enabled environments.*

UNIT– I	**Introduction to Cyberspace and Cybersecurity**
Content	• Introduction to cyberspace and cybersecurity; Evolution of information systems and Internet; Importance of cybersecurity in engineering and society. • Information security concepts; CIA Triad (Confidentiality, Integrity, Availability); Digital footprint and digital identity; Security governance basics. • Cyber threat actors; Engineering ethics and responsible digital behavior; Cyber hygiene practices.
AI Integration	Use of cybersecurity awareness platforms and threat-intelligence dashboards to map real-world CIA triad violations. Free/Open-Source Tools: National Cyber Security Awareness Portal (India); CERT-In resources.

UNIT– II	**Cyber Threats, Cybercrime and Social Engineering**
Content	• Cyber threats and vulnerabilities; Cybercrime and cyber offenders; Malware: Virus, Worm, Trojan, Spyware, Ransomware; Cyberattack lifecycle and common attack methods. • Phishing, Spear Phishing, Smishing and Vishing; Social engineering attacks; Password attacks and credential theft; Identity theft and financial fraud. • Insider threats; AI-enabled cyber threats and deepfakes; Case studies of

	major cyber incidents.
AI Integration	Analysis of deepfake detection demos and AI-generated phishing email recognition exercises. Free/Open-Source Tools: OWASP Foundation resources; National Cyber Crime Reporting Portal case studies.

UNIT– III	**Personal Cyber Safety and Cybersecurity Technologies**
Content	• Password management and Multi-Factor Authentication; Authentication and access control concepts; Safe browsing practices; Email and communication security. • Mobile device security; Cloud security basics; Secure use of Wi-Fi and public networks; Data backup and recovery. • Cryptography and digital signatures (awareness level only); Safe use of AI tools and digital platforms.
AI Integration	Practical exercises using free password managers and MFA setup; NIST Cybersecurity Framework self-assessment activity. Free/Open-Source Tools: NIST Cybersecurity Framework Resources.

UNIT– IV	**Privacy, Cyber Ethics and Cyber Laws**
Content	• Data protection and privacy concepts; Personal data and sensitive information; Cyber ethics and responsible digital behavior. • Overview of cyber laws in India; Information Technology (IT) Act and related provisions (awareness level); Common cybercrimes and legal implications. • Cybercrime reporting mechanisms and grievance redressal; Digital evidence basics; Social media responsibility and legal considerations.
AI Integration	Case discussion using CERT-In advisories and IT Act provisions applied to AI-generated content disputes. Free/Open-Source Tools: National Cyber Crime Reporting Portal; CERT-In.

UNIT– V	**Cyber Hygiene and Secure Digital Practices**
Content	• Cybersecurity risk awareness; Password management and Multi-Factor Authentication; Secure use of email, web, cloud and public networks. • Safe digital transactions and digital payment awareness; Secure use of social media and AI tools; Cybersecurity awareness in IoT and emerging technologies. • Incident awareness and cyber hygiene best practices.
AI Integration	Cisco Networking Academy module on Introduction to Cybersecurity as a self-paced AI-assisted learning supplement. Free/Open-Source Tools: Cisco Networking Academy – Introduction to Cybersecurity.

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Explain fundamental cybersecurity concepts and their importance in digital environments.	L2 (Understand)
CO2	Identify common cyber threats, cyberattacks, and social engineering techniques.	L2 (Understand)
CO3	Apply safe practices for secure use of devices, networks, email, cloud services, and online platforms.	L3 (Apply)
CO4	Demonstrate awareness of cyber ethics, privacy protection, and cyber laws.	L2 (Understand)
CO5	Develop cyber hygiene habits and respond appropriately to common cybersecurity incidents.	L3 (Apply)

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelon Learnin
CO1	L2	2	1	–	–	1	2	–	2	–	1	–	2
CO2	L2	2	2	–	1	1	2	–	2	–	1	–	2
CO3	L3	2	2	1	1	2	2	–	2	1	1	–	3
CO4	L2	1	1	–	–	–	3	1	3	1	2	–	2
CO5	L3	2	2	1	1	2	2	1	2	1	1	1	3

Textbooks :

1. Ajay Singh, *Introduction to Cyber Security: Concepts, Principles, Technologies and Practices*, Latest Edition, Universities Press/Orient BlackSwan, 2019.
2. Nina Godbole and SUnitBelapure, *Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives*, Wiley India, 2011.

Reference Books :

1. Charles J. Brooks, Christopher Grow, Philip Craig and Donald Short, *Cybersecurity Essentials*, 1st Edition, Wiley, 2018.
2. William Stallings and Lawrie Brown, *Computer Security: Principles and Practice*, 4th Edition, Pearson, 2018.
3. Michael E. Whitman and Herbert J. Mattord, *Principles of Information Security*, 6th Edition, Cengage Learning, 2018.

4. Joseph Migga Kizza, *Guide to Computer Network Security*, 2nd Edition, Springer, 2015.
5. Eric Cole, *Cybersecurity for Beginners*, 1st Edition, Wiley, 2023.

Online Resources

- National Cyber Security Awareness Portal (India)
- CERT-In – Indian Computer Emergency Response Team – cert-in.org.in
- National Cyber Crime Reporting Portal – cybercrime.gov.in
- NIST Cybersecurity Framework Resources – nist.gov
- OWASP Foundation – owasp.org
- Cisco Networking Academy – Introduction to Cybersecurity – netacad.com

Recommended Cybersecurity Awareness Resources & Tools

Reporting & Response: CERT-In, National Cyber Crime Reporting Portal (cybercrime.gov.in) – incident reporting and alerts.
Frameworks: NIST Cybersecurity Framework, OWASP – security best practices and awareness.
Authentication & Hygiene: Bitwarden, KeePass, Google Authenticator – password management and Multi-Factor Authentication.
Learning: Cisco Networking Academy (Introduction to Cybersecurity), National Cyber Security Awareness Portal – self-paced awareness training.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26HS0809) APPLIED CHEMISTRY FOR COMPUTER SCIENCE
(Common to all branches)

Prerequisites: Basic Chemistry (Higher Secondary level)

Course Objectives

The objectives of this course are

1 Understand the fundamentals of quantum chemistry including wave-particle duality, quantum numbers, and molecular orbital theory relevant to computing materials.
2. Comprehend the chemistry and electronic properties of semiconductor materials and the chemical processes involved in semiconductor device fabrication.
3. Evaluate electrochemical principles governing batteries, supercapacitors, and fuel cells as energy storage technologies for computing systems.
4. Analyze corrosion mechanisms in electronic devices and design appropriate protective techniques for PCBs, solder joints, and interconnects.
5. Explore the chemistry, synthesis methods, and properties of nanomaterials used in nanoelectronics, quantum computing, and sustainable electronics

Unit – I	**Quantum Chemistry**
Core Content	Fundamentals of quantum chemistry: Black body radiation, photoelectric effect, wave-particle duality, de Broglie wave equation, Heisenberg's uncertainty principle. Fundamentals of Quantum Mechanics: Schrödinger wave equation; significance of Ψ and Ψ^2; quantum numbers. Particle in a one-dimensional box. Molecular orbital theory: Bonding in homonuclear and heteronuclear diatomic molecules; energy level diagrams of O_2 and CO. π-molecular orbitals of Butadiene and Benzene; Calculation of bond order.
AI Integration	Simulation of quantum systems using Python (NumPy, SciPy); visualization of molecular orbitals and energy levels using computational chemistry tools (Google Colab); introduction to quantum computing simulations using Qiskit (introductory level).
Free/Open-Source AI Tools	Python (NumPy, SciPy), Google Colab, Qiskit.

Unit – II	Semiconductor Chemistry

Core Content	Atomic structure and chemical bonding relevant to semiconductors. Band theory: Valence band, conduction band, band gap energy. Intrinsic and extrinsic semiconductors: Silicon, Germanium, Gallium, Arsenide. Doping chemistry: n-type (phosphorus, arsenic) and p-type (boron, aluminium) dopants. Chemical vapour deposition (CVD) for thin film formation; Silicon dioxide (SiO_2) formation and properties. Chemistry of compound semiconductors: GaAs, InP, SiC.
AI Integration	AI-based prediction of semiconductor properties using ML models; process parameter optimization in semiconductor fabrication using machine learning; Python-based band gap prediction using regression models.
Free/Open-Source AI Tools	Python, TensorFlow, Scilab.

Unit – III	Electrochemistry and Energy Storage

Core Content	Electrochemical cells: Galvanic cells, electrode potential, Nernst equation and problems; Potentiometry (redox reactions); Conductometry (strong acid vs. strong base, weak acid vs. strong base). Primary batteries: Zinc-air battery, Sodium-air battery. Secondary batteries: Chemistry of charging and discharging of lead-acid, nickel-cadmium, nickel-metal hydride. Lithium-ion batteries: Construction, working, and applications of $LiCoO_2$. Super capacitors: Definition, classification, and applications. Fuel cells: Definition, basic principle, construction, working, and applications of Solid Oxide Fuel Cells (SOFC) and Direct Methanol Fuel Cells (DMFC). Battery management systems: Overcharge protection and thermal management
AI Integration	AI models for battery health prediction and state-of-charge optimization using scikit-learn; data analysis for energy efficiency in data centres using Python (Pandas); ML-based prediction of electrochemical impedance spectra.
Free/Open-Source AI Tools	Python (Pandas, scikit-learn), Google Colab.

Unit – IV	Chemistry of Corrosion and Protective Techniques

	Concept of Corrosion: Definition, examples, types — chemical corrosion and electrochemical corrosion; Pilling-Bedworth rule. Galvanic corrosion; Concentration cell corrosion; factors affecting corrosion.

Core Content	Deal-Grove model: Thermal oxidation of silicon; passivation layers — silicon dioxide, silicon nitride, and aluminium oxide. Corrosion of printed circuit boards (PCBs): Copper oxidation, tin whiskers; corrosion in solder joints and interconnects. Protective techniques: Proper design, using pure metals, metal alloys, inhibitors; cathodic protection — sacrificial anodic protection, impressed current cathodic protection. Electroplating (nickel) and electroless plating (copper).
AI Integration	Computer vision (OpenCV) for automated defect detection in PCBs; AI-based prediction of corrosion failure in electronic components using TensorFlow; image classification models for identifying corrosion types from microscope images.
Free/Open-Source AI Tools	OpenCV, TensorFlow, Python.

Unit – V	**Chemistry of Nanomaterials**

Core content	Introduction to nanochemistry: Size-dependent properties, quantum confinement. Carbon nanomaterials: Fullerenes, Carbon nanotubes (CNTs), graphene. Synthesis methods: Chemical vapour deposition (CVD), sol-gel process. Quantum dots: Synthesis, properties, applications in displays and quantum computing. Metallic nanoparticles: Preparation and applications of gold, silver, and copper nanoparticles. Applications: Nanoelectronics, interconnects, sensors (optical and biosensors), memory devices. Spintronics materials: Magnetic nanoparticles, topological insulators. Chemistry of quantum computing materials: Superconducting materials (NbTi, Nb_3Sn); Ion trap materials.
AI Integration	AI-based nanomaterial property prediction using machine learning (Python, TensorFlow); simulation of nano-devices and quantum materials using open-source tools; generative AI for exploring nanomaterial design spaces.
Free/Open-Source AI Tools	Python, TensorFlow, open-source materials simulation platforms.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply quantum chemistry principles to analyze molecular systems and explain molecular orbital theory for computing materials.	**L3–L4**
CO2	Analyze semiconductor materials, doping chemistry, and fabrication processes used in modern electronic devices	**L3–L4**
CO3	Evaluate energy storage technologies including batteries, supercapacitors, and fuel cells for computing and data centre	**L3–L4**

	applications.	
CO4	Analyze corrosion mechanisms in electronic devices and recommend appropriate protective techniques for PCBs and interconnects.	**L3–L4**
CO5	Design advanced materials solutions using nanotechnology principles and AI-assisted tools for nanoelectronics and quantum computing.	**L5–L6**

CO – PO Articulation Matrix

Scale: 3 – String, 2 – Moderate, 1 – Slight

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	3	2	–	–	–	–	–	–	2	3	2
CO2	3	3	3	2	3	–	–	–	–	–	–	2	3	3
CO3	2	3	3	2	3	–	2	–	–	–	–	2	3	3
CO4	2	3	2	3	3	–	–	–	–	–	–	1	3	3
CO5	3	2	3	3	3	1	2	1	–	2	–	3	3	3

Text Books:

1. Jain, P. C., & Jain, M. (2020). Engineering Chemistry (17th ed.). Dhanpat Rai Publishing Company.
2. Shika Agarwal. (2026). Engineering Chemistry (5th ed.). Cambridge University Press.

References Books

1. Linden, D., & Reddy, T. B. (Eds.). (2001). Handbook of Batteries (3rd ed.). McGraw-Hill
2. Rao, C. N. R., Müller, A., & Cheetham, A. K. (Eds.). (2007). The Chemistry of Nanomaterials: Synthesis, Properties and Applications (Vols. 1–2). Wiley-VCH.
3. Poole, C. P., & Owens, F. J. (2003). Introduction to Nanotechnology. John Wiley & Sons.
4. Sharma, B. K., & Gaur, H. (2019). Engineering Chemistry. Krishna Prakashan Media.
5. Dara, S. S., & Umare, S. S. (2018). Engineering Chemistry (15th ed.). S. Chand Publishing.

Web References

1. NPTEL Engineering Chemistry: https://nptel.ac.in/
2. IBM Qiskit Learning: https://learning.quantum.ibm.com/
3. Google Colab: https://colab.research.google.com/
4. OpenCV Python Documentation: https://docs.opencv.org/
5. Materials Project (open nanomaterials database): https://materialsproject.org/

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
3	0	2	0	4

(26HS0867) DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS
(Common to all branches)

Prerequisites: Linear Algebra and Calculus

Course Objectives:

The objectives of the course is to

1. *Develop analytical and computational competence in solving ordinary and partial differential equations arising in engineering systems.*
2. *Enable students to interpret vector differential and integral operators in physical and engineering contexts.*
3. *Integrate mathematical modeling, visualization, and simulation using modern AI-assisted computational tools in alignment with NEP 2020 and outcome-based education.*
4. *Apply ODE and PDE solution methods to model real-world engineering phenomena such as circuits, vibrations, heat transfer, and fluid flow.*
5. *Use computational tools for solving, verifying, and visualizing differential equations and vector field problems.*

UNIT– I	**First-Order Ordinary Differential Equations**
Core Content	Classification of ODEs by order, degree, and linearity; linear first-order ODEs and integrating factor method; Bernoulli's equation; exact equations and integrating factors; orthogonal trajectories; Newton's law of cooling (without proof) and natural growth/decay models(without proof); electrical circuit models (RC and RL circuits); engineering applications of first-order ODEs.
AI Integration	Python Sym Pyd solve for symbolic ODE solution and verification; direction field (slope field) visualization using Matplotlib; AI-assisted debugging of integrating factor derivations; using ChatGPT/Copilot to explain solution families and physical interpretations; SciPy solve ivp for numerical ODE solution with initial conditions.
Free/Open-Source AI Tools	Python (SymPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; Wolfram Alpha free tier for solution verification; GNU Octave for ODE routines.

UNIT– II	**Higher-Order Linear Differential Equations**
Core Content	Superposition principle; Wronskian and linear independence; complementary function for constant-coefficient ODEs (distinct real, repeated, and complex roots); particular integral by operator method; variation of parameters; simultaneous linear ODEs; Cauchy's (Euler)

	equation; Legendre's equation; applications to LCR circuits, simple harmonic motion, damped and forced oscillations; mass-spring-dashpot systems.
AI Integration	Python SciPy signal. Sim for simulation of underdamped, critically damped, and overdamped responses; resonance amplitude curve generation; AI-assisted Wronskian computation and verification using SymPy; using AI tools to explain transient vs. steady-state response; resonance and Q-factor interpretation.
Free/Open-Source AI Tools	Python (NumPy, SciPy, SymPy, Matplotlib) on Google Colab; Jupyter Notebook; GNU Octave / MATLAB (optional) for LCR and mechanical system simulation; Wolfram Alpha free tier.

UNIT– III	**Partial Differential Equations**
Core Content	Introduction and classification of PDEs; formation by elimination of arbitrary constants and functions; first-order linear PDEs and Lagrange's method; homogeneous linear PDEs with constant coefficients (CF and PI); standard forms of nonlinear first-order PDEs; second-order PDE applications — Problems on wave equation, heat equation, and Laplace's equation; classification as hyperbolic, parabolic, and elliptic.
AI Integration	Python SymPy for PDE formation and symbolic verification; 3D surface plot visualization of PDE solution families using Matplotlib; SciPy finite-difference simulation of the 1D heat equation with animation; AI-assisted PDE classification using the discriminant; prompting AI to classify mixed-derivative PDEs and student verification of AI reasoning.
Free/Open-Source AI Tools	Python (SymPy, NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; Wolfram Alpha free tier for PDE classification verification; Scilab.

UNIT– IV	**Vector Differentiation**
Core Content	Scalar and vector fields; del (∇) operator; gradient and directional derivative; unit normal to surfaces; angle between surfaces; divergence and physical interpretation; curl and physical interpretation; vector identities; conservative fields and scalar potentials; Laplacian; engineering applications in electromagnetic and fluid mechanics contexts.
AI Integration	Python NumPy and Matplotlib for 2D and 3D gradient, divergence, and curl field visualization (quiver plots, level-surface overlays); SymPy for symbolic verification of vector identities; using AI tools to interpret Maxwell's equations in terms of divergence and curl operators; AI-assisted explanation of irrotational and solenoidal field classification.
Free/Open-Source AI Tools	Python (NumPy, SymPy, Matplotlib, mpl_toolkits) on Google Colab; Jupyter Notebook; SciPy for numerical gradient computation; GNU Octave.

UNIT– V	Vector Integration and Integral Theorems
Core Content	Line integrals over scalar and vector fields; work done and path independence; surface integrals and flux; Green's theorem (without proof) in the plane and its applications; Stokes' theorem (without proof); Divergence theorem (Gauss's theorem) (without proof); applications to area and volume computation; engineering applications in electromagnetism, fluid mechanics, and mass conservation.
AI Integration	Python SciPy integrate for numerical computation of line and surface integrals; 3D flux visualization with colour-coded normal flux using Matplotlib; numerical verification of Green's and Stokes' theorems; AI-assisted physical interpretation of the Divergence theorem in terms of source density; prompting AI to explain connections to Gauss's law and student evaluation of AI explanations.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; SymPy for symbolic integral verification; Wolfram Alpha free tier; 3D plotting tools.

Course Outcomes (COs):

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

CO	Course Outcome Statement	Bloom's Level
CO1	Solve first-order ODEs and model engineering processes using analytical and computational approaches.	**L3 (Apply)**
CO2	Analyze higher-order linear ODEs and interpret solutions in dynamic engineering systems such as circuits and oscillators.	**L4 (Analyse)**
CO3	Formulate and solve first-order and selected higher-order PDEs relevant to engineering models including wave, heat, and Laplace equations.	**L3 (Apply)**
CO4	Interpret and compute gradient, divergence, and curl of scalar and vector fields and explain their physical significance in engineering contexts.	**L4 (Analyse)**
CO5	Apply Green's, Stokes', and Divergence theorems to compute work, circulation, and flux and evaluate engineering field behaviour.	**L5 (Evaluate)**

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L3	Apply	**CO1, CO3**	First-order ODE solution problems; PDE formulation and Lagrange's method exercises.
L4	Analyze	**CO2,**	Higher-order ODE response analysis for LCR

		CO4	circuits; gradient/divergence/curl field interpretation.
L5	Evaluate	**CO5**	Green's/Stokes'/Divergence theorem application for flux and circulation computation.

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	1	-	-	-	1	2	-	-
CO2	3	3	3	2	2	-	-	-	1	2	-	-
CO3	3	2	2	2	1	-	-	-	-	2	-	-
CO4	3	3	3	3	2	-	-	-	1	2	-	-
CO5	3	3	2	3	3	-	1	1	2	2	1	-

CO-PO Correlation Highlights	CO1 through CO4 strongly align with PO1 and PO2 through ODE/PDE solving and vector field analysis. CO5 maps additionally to PO7 and PO8 through engineering field interpretation using integral theorems in electromagnetism and fluid mechanics. PO5 (Modern Tools) is addressed throughout via SymPy, SciPy, and Google Colab-based simulation across all five units, while PO12 (Lifelong Learning) is consistently reinforced through self-directed computational verification practice.

Textbooks :

1. Erwin Kreyszig, *Advanced Engineering Mathematics*, 10th Edition, Wiley, 2018.
2. B. S. Grewal, *Higher Engineering Mathematics*, 44th Edition, Khanna Publishers, 2017.

Reference Books :

1. G. B. Thomas, M. D. Weir and J. Hass, *Thomas' Calculus*, 14th Edition, Pearson, 2018.
2. R. K. Jain and S. R. K. Iyengar, *Advanced Engineering Mathematics*, 5th Edition, Alpha Science International, 2021.
3. D. G. Zill and W. S. Wright, *Advanced Engineering Mathematics*, 6th Edition, Jones & Bartlett Learning, 2018.
4. G. B. Arfken, H. J. Weber and F. E. Harris, *Mathematical Methods for Physicists*, 7th Edition, Academic Press, 2012.
5. I. N. Sneddon, *Elements of Partial Differential Equations*, Dover Publications, 2006.
6. H. F. Davis and A. D. Snider, *Introduction to Vector Analysis*, 7th Edition, Hawkes Publishing, 1995.

Online Resources :

1. NPTEL: Ordinary Differential Equations and Applications – https://nptel.ac.in
2. NPTEL: Partial Differential Equations – https://nptel.ac.in
3. NPTEL: Vector Calculus – https://nptel.ac.in
4. MIT OpenCourseWare – Differential Equations (18.03): https://ocw.mit.edu
5. MIT OpenCourseWare – Multivariable Calculus (18.02): https://ocw.mit.edu

Web Resources & NPTEL Links

1. NPTEL: Ordinary Differential Equations and Applications – https://nptel.ac.in
2. NPTEL: Partial Differential Equations – https://nptel.ac.in
3. NPTEL: Vector Calculus – https://nptel.ac.in
4. MIT OpenCourseWare – Differential Equations (18.03): https://ocw.mit.edu
5. MIT OpenCourseWare – Multivariable Calculus (18.02): https://ocw.mit.edu
6. Python SymPy Documentation: https://docs.sympy.org
7. SciPy Documentation: https://docs.scipy.org
8. Google Colab for Python: https://colab.research.google.com
9. NumPy Documentation: https://numpy.org/doc

Recommended Free & Open-Source AI/Computational Tools

1. Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for ODE/PDE solving, vector field visualization, and numerical integration.
2. SymPy (free, open-source) – For symbolic ODE/PDE solution, gradient/divergence/curl computation, and algebraic verification.
3. Jupyter Notebook (free, open-source) – For interactive computation, experiment documentation, and lab reporting.
4. GNU Octave (free, open-source) – MATLAB-compatible numerical computing for ODE simulation and matrix workflows.
5. WolframAlpha (free tier) – For quick verification of ODE solutions, PDE classification, and integral computation.
6. Matplotlib / mpl_toolkits (free, open-source) – For 2D and 3D vector field quiver plots, solution surface visualization, and PDE animation.
7. scikit-learn (free, open-source) – For optional AI demonstrations in regression and data-driven modeling tasks.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
1	0	4	0	3

(26ME0303) DESIGN THINKING
(Common to all branches)

PREREQISITE: No Prerequisite for the course

COURSE OBJECTIVES

The objectives of this course to

Design Thinking is a human-centred innovation methodology that equips students with a structured approach to understanding users, reframing complex problems, and developing creative solutions. This course integrates Artificial Intelligence tools at each stage of the design cycle—empathise, define, ideate, prototype, and test—enabling students to harness AI not merely as a tool but as a creative collaborator. Students work on real-world projects and submit iterative prototypes.

UNIT – I	**INTRODUCTION TO DESIGN THINKING**

Core Content	• Design Thinking: Definition, origin (Stanford d.school, IDEO), and relevance to engineering • Design Thinking Mindset: Curiosity, empathy, collaboration, experimentation and learning from failure • The 5-Stage Model: Empathise – Define – Ideate – Prototype – Test • Human-centred design principles: Desirability, Feasibility, Viability • Everyday Design Examples: Identifying good and bad designs in everyday products and services • Systems Thinking vs. Linear Thinking: Understanding complexity • Case Studies: Successful design thinking applications (IDEO shopping cart, Stanford Mobile Clinic) • Introduction to Sustainable Design: Considering environmental and social impacts while developing solutions.
AI Integration	**AI Integration:** • AI Tool: ChatGPT / Copilot – use AI to explain design thinking in different domains (healthcare, education, agriculture) • Activity: Use MindMeister (free, open-source alternative: FreeMind) to create a mindmap of the 5-stage process • AI Video Analysis: Watch an IDEO documentary and use AI to summarise key DT principles observed.

UNIT – II	**EMPATHISE AND DEFINE STAGES**

Core Content	• Empathy Methods: Interviews, Observation, Immersion, Shadowing • Stakeholder Identification: Identifying all people affected by a design problem • Creating User Personas: Demographic, psychographic, goals and pain points • Empathy Mapping: Says, Thinks, Does, Feels • Customer Journey Mapping (Basic): Visualising the user's experience across different stages • Point of View (POV) Statement: User + Need + Insight formula • How Might We (HMW) Questions: Reframing problems as opportUNITies • Root Cause Analysis using 5 Whys Technique: Identifying the underlying causes of a problem.
AI Integration	• AI Tool: Miro (free plan) – create digital empathy maps and POV statements collaboratively • AI Tool: ChatGPT – generate diverse user persona profiles for a given problem context • Activity: Interview 5 peers about a campus problem; use AI to analyse interview transcripts and extract themes • AI Tool: Dovetail (free plan) – affinity mapping of qualitative research data.

UNIT – III	IDEATION STAGE
Core Content	• Creative Thinking Techniques: Brainstorming, Brainwriting, SCAMPER, Random Word Association • Crazy 8's Technique: Rapid sketching technique for generating multiple ideas quickly • Idea Mash-up: Combining existing ideas to create innovative concepts • Lateral Thinking: Edward de Bono's Six Thinking Hats • Affinity Diagramming: Grouping ideas by themes • Prioritisation Matrix: Impact vs. Effort, Dot Voting • Concept Selection: Morphological charts and Pugh Matrix • Basic Design Sprint Overview: Understanding rapid idea generation and testing.
AI Integration	• AI Tool: ChatGPT – use 'expand my idea' prompts to generate 20 variations of a concept • AI Tool: IdeaFlip (free) / Miro – digital brainstorming and affinity diagramming • AI Tool: Midjourney (free trial) / DALL-E (free credits) – generate visual concept sketches from text descriptions • Activity: Run a 30-minute AI-augmented brainstorming session and compare idea quality with pure human brainstorming..
UNIT – IV	**PROTOTYPING**
Core	• Principles of Prototyping: Low-fidelity vs. High-fidelity prototypes

Content	• Minimum Viable Product (MVP): Developing essential features of a solution for early testing • Paper Prototyping: Sketching UI and physical models • Storyboarding: Visualising user interaction through sequential sketches. • Digital Wireframing: Introduction to wireframes and mockups • 3D Prototyping Concepts: Introduction to 3D printing and rapid prototyping • Role Play Prototyping: Service design scenarios • Prototype Improvement Checklist: Evaluating usefulness, simplicity and usability before testing.
AI Integration	• AI Tool: Figma (free plan) – create digital wireframes and interactive mockups • AI Tool: Uizard (free plan, AI-powered) – convert hand-drawn sketches to digital wireframes using AI • AI Tool: Tinkercad (free, web-based) – create simple 3D models for physical prototyping • Activity: Build a paper prototype for a chosen problem and photograph for digital documentation.
UNIT – V	**TESTING, ITERATION AND AI IN DESIGN**
Core Content	• User Testing Methods: Think-aloud protocol, A/B testing, Usability testing • Feedback Analysis: Affinity mapping of test feedback • Characteristics of Good Feedback: Specific, constructive and actionable feedback • Iteration Cycle: Incorporating feedback and pivoting • Design Presentation: Pitching to stakeholders (elevator pitch, demo day format) • AI in Design: Overview of AI-assisted design (generative design, co-creation with AI) • Introduction to Responsible AI: Human oversight, ethical use and verification of AI-generated outputs. • Innovation and Entrepreneurship (Basic): Understanding how Design Thinking supports innovation and product development.
AI Integration	• AI Tool: Maze (free plan) – conduct remote usability tests on Figma prototypes • AI Tool: ChatGPT – analyse collected user feedback and generate insights and improvement suggestions • Activity: Present a final prototype to a panel; use AI to prepare pitch slides in Canva • Reflection: Create a Design Thinking journal documenting each stage with AI tool usage notes

COURSE OUTCOMES (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and understand the philosophy, principles, and stages of the Design Thinking process and its relevance to human-centred problem solving.	L1 – Remember / L2 – Understand
CO2	Apply empathy tools (interviews, observation, personas) and problem-framing techniques to real-world engineering scenarios.	L3 – Apply
CO3	Analyse user needs and problem statements using the 'How Might We' framework and AI-powered insight generation tools.	L4 – Analyse
CO4	Evaluate and select the most viable prototype solution through structured testing, user feedback, and AI-assisted evaluation rubrics.	L5 – Evaluate
CO5	Create innovative prototypes and pitch solutions to identified problems using AI ideation tools and digital prototyping platforms.	L6 – Create

CO-PO MAPPING MATRIX

Scale: 3 = High, 2 = Medium, 1 = Low, '-' = No correlation

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	2	1	1	2	1	1	2	2	1	2	1	1
CO2	1	2	3	2	2	3	1	1	3	2	2	2	2	2
CO3	2	3	3	2	3	3	1	1	2	2	2	3	2	2
CO4	2	3	3	3	3	2	1	2	2	2	2	3	2	2
CO5	2	3	3	3	3	2	2	2	3	3	3	3	3	3

LEARNING RESOURCES

TEXTBOOKS

1. Brown, Tim. Change, *Design: How Design Thinking Transforms Organizations and Inspires Innovation*. HarperBusiness, New York, 2019.
2. Lewrick, Michael, Link, Patrick & Leifer, Larry, *The Design Thinking Playbook*, Wiley, New Jersey, 2018.

REFERENCES

1. IDEO.org, *The Field Guide to Human-Centered Design*, IDEO.org, San Francisco, 2015.
2. Kelley, Tom & Kelley, David, *Creative Confidence: Unleashing the Creative Potential Within Us All*, Crown Business, 2013.
3. Liedtka, Jeanne & Ogilvie, Tim, *Designing for Growth: A Design Thinking Tool Kit for Managers*, Columbia Business School Press, 2011.

4. Plattner, Hasso, Meinel, Christoph & Leifer, Larry (Eds.), *Design Thinking Research*, Springer, 2018.
5. Kumar, Vijay, *101 Design Methods: A Structured Approach for Driving Innovation*, Wiley, 2012.
6. Stickdorn, Marc & Schneider, Jakob, *This Is Service Design Thinking (Expanded ed.)*, BIS Publishers, 2021.

WEB RESOURCES

1. Stanford d.school Resources: dschool.stanford.edu/resources
2. IDEO Design Kit: www.designkit.org/methods
3. MIT OpenCourseWare – Design Thinking: ocw.mit.edu
4. Interaction Design Foundation: www.interaction-design.org
5. NPTEL Design Thinking: swayam.gov.in

YouTube / Video Resources

1. YouTube: Stanford d.school – Design Thinking Bootcamp – www.youtube.com/@stanforddschool
2. YouTube: IDEO U – www.youtube.com/@ideou
3. YouTube: Crash Course – Design Thinking Series – www.youtube.com/@crashcourse
4. YouTube: AJ&Smart – Design Sprint tutorials – www.youtube.com/@AJSmart
5. YouTube: NPTEL Design Thinking – www.youtube.com/c/nptel

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0503) DATA STRUCTURES & EFFICIENT PROBLEM SOLVING
(Common to CSE & Allied branches)

L	T	P	C
3	0	0	3

Course Objectives :

The objectives of the course is to

1. *Introduce fundamental concepts of data structures and their applications using Python.*
2. *Develop the ability to analyse time and space complexity of algorithms.*
3. *Implement linear and non-linear data structures using Python.*
4. *Apply appropriate data structures for efficient problem solving.*
5. *Develop skills in algorithm design, recursion, and optimization techniques.*

UNIT– I	**Introduction, Algorithm Analysis, And Linear Structures**
Content	Introduction to data structures; Abstract Data Types (ADTs); algorithm analysis, time and space complexity; Big-O, Big-Omega, Big-Theta notations. Lists in Python: operations on lists; static vs. dynamic memory allocation. Searching techniques: Linear Search, Binary Search, Interpolation Search analysis and comparison. Sorting techniques: Bubble Sort, Selection Sort, Insertion Sort , algorithms, trace, and complexity analysis. **Source:** TB1: Chapters 1–3 (Vasudevan et al.); TB2: Chapters 1, 5 (Goodrich et al.)
AI Integration	Use VisuAlgo and Algorithm Visualizer to animate sorting and searching; use ChatGPT to explain Big-O complexity with concrete examples; use Python Tutor to trace memory representation of data structures.

UNIT– II	**Stacks And Queues**
Content	Stacks: LIFO principle, push, pop, peek; implementation using Python lists; applications , expression evaluation (infix to postfix, postfix evaluation), parenthesis checking. Queues: FIFO principle, enqueue, dequeue; types , Circular Queue, Priority Queue, Deque; implementation using Python lists and linked structures. Applications of queues: BFS simulation, scheduling, real-world analogies. **Source:** TB1: Chapters 4–5 (Vasudevan et al.); TB2: Chapter 6 (Goodrich et al.)
AI Integration	Use VisuAlgo to animate stack and queue operations; use ChatGPT to generate edge cases (overflow, underflow, circular behaviour); use GitHub Copilot to implement expression evaluation.

UNIT– III	Linked Lists
Content	Singly Linked List: node structure, insertion at head/tail/position, deletion by value/posiion, traversal, reversal. Doubly Linked List: forward and backward traversal, insertion, deletion; Circular Linked List: traversal and operations. Applications: polynomial representation and addition, browser history simulation (doubly linked list), symbol table implementation. **Source:** TB1: Chapter 6 (Vasudevan et al.); TB2: Chapter 7 (Goodrich et al.)
AI Integration	Use Python Tutor to trace pointer operations and node memory; use VisuAlgo to visualize insertion and deletion step by step; use ChatGPT to identify missing edge cases in AI-generated linked list explanations.

UNIT– IV	Trees and heaps
Content	Trees: terminology, properties; Binary Trees representation, traversals (inorder, preorder, postorder, level order). Binary Search Trees (BST): insertion, deletion, search; applications autocomplete, range queries, dictionary lookup. Heaps: Min-Heap and Max-Heap, array representation, heapify, Build-Heap; Heap Sort , algorithm and complexity. **Source:** TB1: Chapters 7–8 (Vasudevan et al.); TB2: Chapters 8, 9 (Goodrich et al.)
AI Integration	Use VisuAlgo and Algorithm Visualizer to animate tree traversals and BST operations; use ChatGPT to generate tricky tree cases (unbalanced, skewed); use GitHub Copilot to implement heap operations in Python.

UNIT– V	Graphs And Hashing
Content	Graphs: terminology, types (directed, undirected, weighted); representation adjacency matrix and adjacency list. Graph traversals: BFS (queue-based) and DFS (stack/recursion-based); applications shortest path, cycle detection, social networks. Hashing: hash functions, properties of a good hash function; collision resolution chaining (separate chaining), open addressing (linear probing, quadratic probing, double hashing); load factor. **Source:** TB1: Chapters 9–10 (Vasudevan et al.); TB2: Chapters 14, 10 (Goodrich et al.)
AI Integration	Use NetworkX in Python and Gephi to build and traverse graphs; use VisuAlgo to animate BFS and DFS; use ChatGPT to explain real-world graph applications; use GitHub Copilot to implement hash tables with

	collision resolution.

Suggested Tools: NetworkX, Gephi, VisuAlgo, ChatGPT, GitHub Copilot

Course Outcomes (Cos):

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Apply basic data structure concepts and analyse algorithm complexity using Python. (L3)	L3 (Apply)
CO2	Implement searching and sorting techniques for efficient data processing. (L3)	L3 (Apply)
CO3	Develop programs using linear data structures — stacks, queues, and linked lists — in Python. (L4)	L4 (Analyse)
CO4	Design and implement non-linear data structures including trees and heaps using Python. (L4)	L4 (Analyse)
CO5	Apply graph traversal and hashing techniques to solve real-world computational problems. (L4)	L4 (Analyse)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define ADT, Big-O notation; describe stack, queue, tree, graph terminology.
L3	Apply	**CO1–CO3**	Implement stacks, queues, and linked list operations; trace sorting algorithms.
L4	Analyse	**CO3–CO5**	Compare data structures; trace BFS vs. DFS; analyse BST vs. Heap operations.
L5	Evaluate	**CO4, CO5**	Evaluate hash collision strategies; assess Heap Sort vs. Merge Sort.
L6	Create	**CO4, CO5**	Design a Python solution using appropriate data structure for a domain problem.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO 1	L3	**3**	**2**	1	1	1	1	–	–	–	1	–
CO 2	L3	**3**	**3**	**2**	1	1	–	–	–	–	1	–
CO 3	L4	**3**	**3**	**2**	**2**	**2**	–	–	–	–	1	–
CO 4	L4	**3**	**3**	**3**	**2**	**2**	–	–	–	1	**2**	–
CO 5	L3	**3**	**3**	**2**	1	**2**	–	–	–	–	1	–

CO-PO Correlation Highlights	Strong PO1 and PO2 through algorithm complexity analysis and data structure design. PO3 (Design/Development) through tree, heap, graph, and hash table design in CO4 and CO5. PO4 (Investigation) through experimental complexity measurement and traversal analysis using VisuAlgo.

Textbooks :

1. Shriram K. Vasudevan, Abhishek S. Nagarajan & Karthick Nanmaran, *Data Structures Using Python*, Wiley India.
2. Michael T. Goodrich, Roberto Tamassia & Michael H. Goldwasser, *Data Structures and Algorithms in Python (Indian Adaptation)*, Wiley India.

Reference Books :

1. Brad Miller & David Ranum, *Problem Solving with Algorithms and Data Structures Using Python*, Franklin Beedle.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, Pearson Education.

Online Resources :

- NPTEL – Data Structures and Algorithms – nptel.ac.in
- Visu Algo – visualgo.net
- Python Tutor – pythontutor.com
- Network X – networkx.org
- Geeks for Geeks – geeksforgeeks.org
- Algorithm Visualizer – algorithm-visualizer.org
-

Recommended Free & Open-Source Data Structures / AI Tools

Language & Environment: Python, Jupyter Notebook – implementation and tracing.
Visualization: VisuAlgo, Algorithm Visualizer, Python Tutor – animate sorting, traversal and memory operations.
Graph Tools: NetworkX, Gephi – graph build and traversal.
AI Assistants: ChatGPT, GitHub Copilot – complexity explanation, edge-case generation and implementation.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
1	0	4	0	3

(26ME0301) ENGINEERING GRAPHICS
(Common to all branches)

PREREQUISITES: Basic Geometry, Fundamental Python Programming

COURSE OBJECTIVES

The objectives of this course is to

1. Familiarise with the BIS conventions, geometric construction standards, and engineering scales
2. Develop visualization skills to interpret and project points, lines, and regular planes into 2D drawing sheets
3. Impartfoundational knowledge on projecting regular 3D solids and reading directional geometric orientations.
4. Enable the spatial reconstruction of sliced solids and flat unfolded developments of various structural shells.
5. Introduce CAD tools combined with generative AI software to construct multi-view isometric representations from 3D models.

UNIT – I	**INTRODUCTION TO ENGINEERING GRAPHICS, CURVES, AND SCALES**
Core Content	**Introduction:** Principles of Engineering Graphics, signboards, BIS conventions, and lettering, Construction of Polygons. **Curves:** Conic sections: Ellipse, Parabola, and Hyperbola using eccentricity/general methods. Cycloidal curves, Involutes for regular polygons and Circle. **Scales:** Plain, Diagonal, and Vernier scales. **Hands-on Experience:** Manual draft generation of an ellipse and diagonal scale, paired with a software verification loop.
AI Integration	**AI Integration Module**: AI prompt engineering for generating geometric profiles; algorithmic generation of conic curves using mathematical code blocks; parsing hand-drawn geometric sketches into clean vector shapes using Computer Vision (CV) edge detection (Canny Edge/Hough Transform).
UNIT – II	**ORTHOGRAPHIC PROJECTIONS OF POINTS, LINES**
Core Content	Principles of orthographic projections, projection of points in all four quadrants. Projection of straight lines inclined to one or both reference planes, traces of lines, determination of true lengths and true inclinations.

	Projections of Planes: regular planes Perpendicular to both reference planes, parallel to one reference plane and inclined to the other reference plane; plane inclined to both the reference planes. **Auxiliary Projections:** Projection of Points, straight lines simple problems only **Hands-On Experience & Application** **Project**: Construct a transformation script where a user inputting 2D projection endpoints automatically outputs the 3D line representation, true spatial length, and inclination angles without manual geometric tracking. **Application**: Calibration setups for stereoscopic robotic arms tracking trajectories across multiple camera planes.
AI Integration	Machine learning pipelines for 3D coordinate estimation from 2D views; multi-view structural tracking; training neural models to predict true line length and orientation angles based solely on coordinate projections.
UNIT – III	PROJECTION OF REGULAR PLANES, SOLIDS
Core Content	**Projection of Regular Solids**: Types of solids: Polyhedra and Solids of revolution. Projections of solids in simple positions: Axis perpendicular to horizontal plane, Axis perpendicular to vertical plane and Axis parallel to both the reference planes, Projection of Solids with axis inclined to one reference plane and parallel to another plane, Solids with axis inclined to both reference planes. **Hands-On Experience & Application** • **Project**: Set up a classification notebook using scikit-learn or a simple convolutional model to automatically identify the type of solid (e.g., hexagonal prism vs. cone) present within an orthographic engineering layout image. • **Application**: Automated automated component sorting inside factory staging bays and conveyor streams.
AI Integration	Object recognition models (e.g., YOLO / MobileNet) trained to identify standard solid geometries; automated multiview projection generation from 3D CAD meshes using programmatic render layers.
UNIT– IV	**SECTIONS OF SOLIDS, DEVELOPMENT OF SURFACES, AND AI GENERATIVE DESIGN**
Core Content	**Sections of Solids**: Parallel, Perpendicular and inclined section planes, Sectional views and True shape of section, Sections of solids in simple position only. **Development of Surfaces:** Methods of Development: Parallel line development and radial line development. Development of a cube, prism, cylinder, pyramid and cone.

	Hands-On Experience & Application • **Project**: Use a nesting optimization script (e.g., using genetic algorithms or standard heuristics) to arrange several custom flat-surface solid developments onto a finite raw sheet grid, actively minimizing material margins and cut lines. **Application**: Sheet metal stamping pattern automation in automotive and aerospace chassis manufacturing
AI Integration	Generative Adversarial Networks (GANs) for generating flat-pattern configurations; algorithmic optimization of surface layouts to minimize sheet manufacturing scrap; predictive tool path generation using physics-guided heuristic networks
UNIT – V	ISOMETRIC PROJECTIONS, CONVERSIONS, AND 2D-TO-3D DEEP RECONSTRUCTION
Core Content	Principles of isometric projection, isometric scale, isometric views/projections of planes, simple, compound solids and other 3D objects. Conversion of isometric views to orthographic views and vice-versa. **Computer graphics**: Creating 2D&3D drawings of objects including PCB and Transformations using Auto CAD (*Not for end examination*). **Hands-On Experience & Application** • **Project**: Use a pretrained depth estimation or pixel-to-voxel neural model to parse a set of front, top, and side drawings, compiling them directly into an interactive 3D point cloud or boundary surface rendering. • **Application**: Instantly generating 3D printable mechanical files directly from archival 2D layout drafting logs.
AI Integration	2D-to-3D depth reconstruction using Neural Radiance Fields (NeRF) or specialized generative models; auto-generation of 3D CAD models from simple multi-view engineering layout sketches.

COURSE OUTCOMES (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Construct engineering curves and scales using traditional drafting geometry validated by automated algorithmic paths.	L3 (Apply)
CO2	Solve multi-planar projection instances of linear points, segments, and standard regular planar silhouettes.	L3 (Apply)
CO3	Render projections of standard 3D solids and identify variations using structural boundary algorithms.	L4 (Analyze)

CO4	Draft complex cross-sectional geometries and mapped surface extensions of sliced geometric forms.	L3 (Apply)
CO5	Produce composite isometric layouts alongside generative text-to-CAD system alternatives for dynamic prototyping.	L6(Create)

CO – PO Articulation Matrix

Scale: **1** (Low/Slight), **2** (Medium/Moderate), and **3** (High/Substantial)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	1	-	-	-	-	-	-	2	-	1
CO2	3	3	2	-	-	-	-	-	-	2	-	1
CO3	3	3	2	1	-	-	-	-	-	2	-	1
CO4	3	3	3	1	2	-	-	-	-	2	-	2
CO5	3	2	2	-	3	-	-	-	1	3	-	2

LEARNING RESOURCES

TEXT BOOKS

1. N. D. Bhatt, *Engineering Drawing,* Charotar Publishing House, 2016.
2. Dhananjay Jolhe, *Engineering Drawing with an Introduction to AutoCAD,* TataMcGraw Hill, 2017.

REFERENCES

1. Pradeep Jain, A.P. Gautam, Abhishek Bhargava, *Engineering Graphics and Design*, Khanna Publishing
2. K.L. Narayana and P. Kannaiah, *Engineering Drawing,* Tata McGraw Hill, 2013.
3. M.B.Shah and B.C. Rana, *Engineering Drawing,* Pearson Education Inc,2009.

VIDEO LINKS & ONLINE LECTURES

- **Classical Drafting Fundamentals**: NPTEL Engineering Graphical Design Course by IIT Kharagpur
- **Computer Vision and Processing**: OpenCV Official Python Bootcamps and Tutorials
- **Generative Design and Modeling Insights**: MIT 6.S192: Deep Generative Modeling Platform

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26ME0302) ENGINEERING WORKSHOP
(Common to all branches)

PREREQUISITES: Basic Computing Elements

COURSE OBJECTIVES

The objectives of this course is to

1. *Create awareness on safety precautions required at workplace.*
2. *Master fundamental hand tools and traditional manufacturing craftsmanship.*
3. *Practice on manufacturing of components using workshop trades including fitting, Sheet metal and carpentry, foundry and welding.*
4. *Import basic electrical engineering knowledge for House Wiring Practice.*
5. *Demonstrate fundamentals of foundry and welding.*

SYLLABUS

Exp – 1	**SAFETY PRACTICES & PRECAUTIONS**
Content	**Demonstrate the necessity of** Workshop hazards, various Personal Protective Equipment and emergency protocols.
AI Integration	**AI Integration:** Computer vision for safety compliance.
Exp – 2	**WOOD WORKING**
Content	Introduction to different types of woods, carpentry tools, timber joints a) Half – Lap joint b) Mortise and Ten on joint c) Corner Dovetail joint or Bridle joint
Exp – 3	**FITTING**
Content	Familiarity with different types of tools used in fitting and perform fitting exercises like a) V-fit b) Dovetail fit c) Semi-circular fit
Exp – 4	**SHEET METAL WORKING**
Content	Familiarity with different types of tools used in sheet metal working, Developments of following sheet metal job from GI sheets. a) Tapered tray b) Conical funnel c) Elbow pipe d) Brazing
Exp – 5	**ELECTRICAL WIRING**
Content	Introduction to electrical tools, safety precautions, wiring materials. Familiarity with different types of basic electrical circuits and make the following connections. a) Parallel and series b) Two-way switch c) God own lighting

	d) Tube light e) Three phase motor f) Soldering of wires
Exp – 6	**PLUMBING**
Content	Introduction to plumbing tools, pipe fittings, thread cutting, making a simple PVC pipe Joint. Preparation of Pipe joints with coupling for same diameter and with reducer for different diameters
Exp – 7	**WELDING SHOP**
Content	Demonstration and practice on Arc Welding and Gas welding. Preparation of a simple Lap joint/Butt joint.
Exp – 8	**FOUNDRY TRADE**
Content	Demonstration and practice on Moulding tools and processes, Preparation of Green Sand Moulds for a simple Patterns.
Exp – 9	**INTRODUCTION TO 3D PRINTING**
Content	Demonstrate the core concepts of 3D printing, Hardware, software, Digital file and 3D printing materials.Familiarise with 3D Printing Process of complex objects.
Exp – 10	**COMPUTER COMPONENTS**
Content	Disassemble and reassemble a desktop computer to understand the role of each component (CPU, motherboard, RAM, HDD/SSD, GPU, etc.). **Hardware Troubleshooting & Peripheral Installation:** Simulate common hardware issues (e.g., RAM failure, CPU overheating) and identify the procedures to resolving them. **Peripheral Installation:** Install and configure peripheral devices such as printers, scanners, and external hard drives. **Network Setup and Configuration:** Set up a small LAN network with routers, switches, and computers. Configure IP addresses, subnet masks, and basic network services.
Exp-11	**DISASSEMBLY AND ASSEMBLY OF EQUIPMENT SUCH AS BICYCLE, GEAR BOX, BRAKING SYSTEM:**
Content	**F**amiliarise with disassembly and assembly process

AI Integration Module: Smart energy monitoring using IoT current sensors; building machine learning load forecasting models; training anomaly detection systems to automatically flag electrical arc or leakage patterns. Develop predictive load-monitoring models that detect faulty configurations or phantom energy drains.

ADDITIONAL EXPERIMENTS
1. Design and modelling of engineering components using CAD software. 2. Development of basic geometric models using a 3D printing machine 3. Fabrication of wooden models using CNC wood working machine. 4. Demonstration of PCB design and printing. 5. Fabrication of components/models using a CO_2 laser cutting machine. 6. Fabrication of name plates using a vinyl sheet cutting machine.

COURSE OUTCOMES

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Recognize the importance of safety precautions while working in the workshop..	L3 (Apply)
CO2	Identify different measuring, marking, cutting tools used in the workshop and their operational Capabilities.	L4 (Analyse)
CO3	Fabricate precision wood and metal fitting joints within standard specifications.	L4 (Analyse)
CO4	Understand Wire standard, safe residential switches, lighting circuits, and basic appliances.	L5 (Evaluate)
CO5	Identify different tools used in foundry and welding sections.	L5 (Evaluate)

Engineering Workshop CO-PO Matrix												
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	2	-	-	-	-	-	-	2
CO2	3	2	2	1	2	-	-	-	-	-	-	1
CO3	3	1	-	-	1	2	-	-	-	-	-	2
CO4	-	-	-	-	-	-	-	1	3	-	1	-
CO5	-	-	-	-	-	-	-	-	-	3	-	1
1 (Low/Slight), 2 (Medium/Moderate), and 3 (High/Substantial)												

LEARNING RESOURCES

TEXTBOOKS

1. S.K. Hajra Choudhury, *Workship Technology,* Media Promoters, *Vol I & II*.
2. B.S. Raghuwanshi, *A Course in Workshop Technology Vol I. & II,* Dhanpath Rai & Co., 2015 & 2017.
3. Giacomo Veneri and Antonio Capasso, *Hands-On Industrial Internet of Things,* Packt Publishing.
4. Felix W.; Independently Published, 2019. *Workshop Processes, Practices and Materials;* Bruce J. Black, Routledge publishers, 5th Edn. 2015
5. Bruce J. Black, *Workshop Processes, Practices and Materials*, Routledge publishers, 5th Edn. 2015.
6. Soni P.M. & Upadhyay P.A, *Wiring Estimating, Costing and Contracting*, Atul Prakashan, 2021-22.

REFERENCES

1. P.N. Rao, *Manufacturing Technology (Foundry, Forming and Welding)*, McGraw Hill.

2. R. Keith Mobley and Butterworth-Heinemann, *An Introduction to Predictive Maintenance.*

VIDEO LINKS & ONLINE LECTURES

- **Classical Shopfloor Operations**: NPTEL Workshop Practices Video Series by IIT Kharagpur
- **Industrial AI & Predictive Analytics**: Predictive Maintenance Models in Python Tutorial
- **Smart Sensors & Electronics Fabrication**: MIT 6.131: Power Electronics and Smart Grid Interfaces

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26HS0810) APPLIED CHEMISTRY FOR COMPUTER SCIENCE LAB
(Common to all branches)

COURSE OBJECTIVES

The objectives of this course

- Perform classical wet chemical analyses (redox, acid-base) on industrially relevant systems including battery electrolytes and iron solutions.
- Determine corrosion rates and understand the influence of environmental factors using standard electrochemical and gravimetric methods.
- Apply instrumental methods—conductometry, potentiometry, spectrophotometry, and IR spectrophotometry—for quantitative and qualitative chemical analysis.
- Synthesise a polymer (Bakelite) and relate its properties to structure and industrial relevance for electronics applications.

List of Experiments:

1. Estimation of Ferrous Iron (Fe^{2+}) using Potassium Permanganate ($KMnO_4$) — Redox Titration
2. Determination of Strength of Acid (H_2SO_4) in Battery Using Sodium Hydroxide (NaOH)
3. Determination of Corrosion
4. Determination of Corrosion by Environment
5. Conductometric Titration of Strong Acid vs Strong Base (HCl vs NaOH)
6. Conductometric Titration of Weak Acid vs Strong Base (CH_3COOH vs NaOH)
7. Potentiometry (Redox Titration) — Estimation of Iron (Fe^{2+}) Using Potassium Dichromate ($K_2Cr_2O_7$)
8. Spectrophotometry — Determination of Wavelength (λ) and Verification of Beer-Lambert's Law
9. Identification of Functional Groups in Organic Compound by Infra-Red (IR) Spectrum
10. Preparation of Polymer (Phenol-Formaldehyde — Bakelite).
11. Preparation of Zinc Oxide (ZnO) Nanoparticles by the Sol-Gel Method
12. Determination of Band Gap Energy of a Semiconductor.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Perform redox and conductometric titrations to estimate concentrations of analytes in solution.	**L3 (Apply)**
CO2	Determine corrosion rates and analyze environmental factors affecting corrosion of metals.	**L3–L4**
CO3	Apply potentiometric and spectrophotometric techniques for quantitative chemical analysis.	**L3 (Apply)**
CO4	Identify functional groups in organic compounds using IR spectroscopy.	**L3 (Apply)**
CO5	Synthesize a polymer and use AI/Python-based data analysis for predictive modelling of material behaviour.	**L3–L4**

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	2	-	1	2	-	-	-	-	-	3	2
CO2	2	-	-	2	-	2	1	-	-	-	-	-	2	3
CO3	2	-	-	3	-	3	2	-	-	-	-	-	2	3
CO4	2	-	-	2	-	2	1	-	-	-	-	-	3	2
CO5	1	-	-	2	3	2	2	-	-	-	-	-	2	3

Textbooks:

1.Jain, P. C., & Jain, M. (2020). Engineering Chemistry (17th ed.). Dhanpat Rai Publishing.

2.Shika Agarwal (2026). Engineering Chemistry (5th ed.). Cambridge University Press.

Reference Books:

1.Dara, S. S., & Umare, S. S. (2018). Engineering Chemistry (15th ed.). S. Chand Publishing.

Online Resources:

- IEEE Xplore Digital Library
- Semiconductor Industry Association resources
- NPTEL Engineering Chemistry

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	2	1

(26CS0504) DATA STRUCTURES & EFFICIENT PROBLEM SOLVING LAB
(Common to all branches)

Course Objectives :

The objectives of the course is to

1. *Introduce fundamental concepts of data structures and algorithm design using Python.*
2. *Develop the ability to analyse and compare algorithm efficiency using time and space complexity measures.*
3. *Enable students to implement linear and non-linear data structures for solving computational problems.*
4. *Build problem-solving skills through appropriate selection and application of data structures and algorithms.*
5. *Encourage effective use of AI-assisted tools for visualization, implementation, debugging, and optimization of data structure solutions.*

List of Experiments

MODULE I: Introduction Algorithm Analysis & Linear Structures

Topics: Introduction to Data Structures; Abstract Data Types (ADTs); algorithm analysis; time and space complexity; Big-O, Big-Omega, Big-Theta; lists in Python; searching; sorting.

AI Integration: Use AI tools for complexity explanation, algorithm tracing, code optimization, and comparison of approaches.

Suggested Tools: Python, Jupyter Notebook, VisuAlgo, Python Tutor, Algorithm Visualizer

Experiments:

1. **Experiment 1: Complexity Analysis of Python Programs ,** Implement simple programs and analyse time complexity using Big-O notation; compare multiple implementations using AI-generated complexity explanations.
2. **Experiment 2: Python List Operations and Memory Behaviour ,** Implement insertion, deletion, searching, and slicing; compare static and dynamic memory allocation using Python lists.
3. **Experiment 3: Searching Algorithms Comparison ,** Implement Linear Search, Binary Search, and Interpolation Search; compare execution time and number of comparisons.
4. **Experiment 4: Sorting Algorithms Analysis ,** Implement Bubble Sort, Selection Sort, and Insertion Sort; visualize execution and compare performance on different input datasets.

MODULE II: Stacks And Queues

Topics: Stacks; expression evaluation; parenthesis checking; queues; Circular Queue; Priority Queue; Deque; queue applications.

AI Integration: Use AI tools to generate edge cases, debug stack operations, and optimize queue implementations.

Suggested Tools: Python, VisuAlgo, GitHub Copilot, Python Tutor

Experiments:

1. **Experiment 5: Stack Implementation and Applications -** Implement stack operations using Python lists and solve parenthesis-matching problems.
2. **Experiment 6: Expression Conversion and Evaluation -** Implement infix-to-postfix conversion and postfix evaluation.
3. **Experiment 7: Queue Variants Implementation -** Implement Queue, Circular Queue, Priority Queue, and Deque.
4. **Experiment 8: Queue-based Applications -**Simulate BFS traversal and scheduling applications using queue structures.

MODULE III: Linked Lists

Topics: Singly Linked List; Doubly Linked List; Circular Linked List; applications.

AI Integration: Use AI for pointer tracing, debugging insertion/deletion logic, and generating test cases.

Suggested Tools: Python Tutor, VisuAlgo, Jupyter Notebook

Experiments:

1. **Experiment 9: Singly Linked List Operations -** Implement insertion, deletion, traversal, and reversal operations.
2. **Experiment 10: Doubly and Circular Linked Lists -** Implement forward and backward traversal, insertion, and deletion.
3. **Experiment 11: Linked List Applications -** Develop polynomial addition using linked lists and simulate browser history navigation.

MODULE IV: Trees And Heaps

Topics: Binary Trees; tree traversals; Binary Search Tree (BST); heaps; Heap Sort.

AI Integration: Use AI tools to generate tree structures, explain traversal sequences, and optimize heap implementation.

Suggested Tools: Python, VisuAlgo, Algorithm Visualizer, GitHub Copilot

Experiments:

1. **Experiment 12: Binary Tree Construction and Traversals -** Implement inorder, preorder, postorder, and level-order traversals.
2. **Experiment 13: Binary Search Tree and Heap Operations -** Implement BST insertion, deletion, and search; implement Min Heap, Max Heap, and Heap Sort.

MODULE V: Graphs And Hashing

Topics: Graphs; BFS; DFS; hash functions; collision resolution.

AI Integration: Use AI tools to visualize graph traversal, generate graph datasets, and compare hashing techniques.

Suggested Tools: Python, NetworkX, Gephi, VisuAlgo

Experiments:

1. **Experiment 14: Graph Representation and Traversal ,** Represent graphs using adjacency matrix and adjacency list; implement BFS and DFS.

2. **Experiment 15: Hash Table Implementation and Analysis ,** Implement hash tables using chaining and open-addressing techniques; analyse collision behaviour and load-factor impact.

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Apply data structure concepts and analyse algorithm complexity using asymptotic notations and Python implementations.	L3 (Apply)
CO2	Implement and evaluate searching and sorting algorithms for efficient data processing.	L3 (Apply)
CO3	Develop applications using linear data structures including stacks, queues, and linked lists in Python.	L4 (Analyze)
CO4	Design and implement non-linear data structures including trees and heaps for problem solving.	L4 (Analyze)
CO5	Apply graph traversal and hashing techniques to develop efficient computational solutions and analyse their performance.	L4 (Analyze)

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	Bloom's Level	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	2	3	–	–	–	–	1	–	2	3	2
CO2	L3	3	3	2	2	3	–	–	–	–	1	1	2	3	2
CO3	L4	3	3	3	2	3	–	–	–	1	1	1	2	3	2
CO4	L4	3	3	3	2	3	–	–	–	1	1	1	2	3	2
CO5	L4	3	3	3	3	3	–	–	–	1	1	1	2	3	2

Textbooks :

1. Shriram K. Vasudevan, Abhishek S. Nagarajan and Karthick Nanmaran, *Data Structures Using Python*, 1st Edition, Wiley India, 2019.
2. Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser, *Data Structures and Algorithms in Python (Indian Adaptation)*, 1st Indian Edition, Wiley India, 2017.

Reference Books :

1. Brad Miller and David Ranum, *Problem Solving with Algorithms and Data Structures Using Python*, 1st Edition, Franklin, Beedle & Associates, 2011.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, 2nd Edition, Pearson Education, 2007.
3. Narasimha Karumanchi, *Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles*, 5th Edition, CareerMonk Publications, 2017.

4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, 4th Edition, MIT Press, 2022.
5. Aditya Bhargava, *Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People*, 1st Edition, Manning Publications, 2016.

Online Learning Resources:

1. NPTEL – Data Structures and Algorithms – nptel.ac.in
2. VisuAlgo – visualgo.net
3. Python Tutor – pythontutor.com
4. Algorithm Visualizer – algorithm-visualizer.org
5. NetworkX Documentation – networkx.org
6. GeeksforGeeks – Data Structures – geeksforgeeks.org
7. Python Documentation – docs.python.org

Recommended Free & Open-Source Data Structures / AI Tools

Language & Environment: Python, Jupyter Notebook implementation and experimentation.
Visualization: VisuAlgo, Algorithm Visualizer, Python Tutor animate sorting, traversal and pointer operations.
Graph Tools: NetworkX, Gephi – graph construction and traversal.
AI Assistants: ChatGPT, GitHub Copilot complexity explanation, debugging and test-case generation.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	1	0.5

(26HS0824) HEALTH & WELLNESS, YOGA AND SPORTS
(Common to all branches)

Course Objectives
The objectives of this course are to

1. Create awareness about holistic health, wellness and preventive healthcare among students.
2. Train students in yogic practices (asanas, pranayama, meditation) for physical and mental well-being.
3. Enable students to design evidence-based personal fitness and nutrition plans.
4. Develop sportsmanship, team spirit, fair play and physical fitness through indoor and outdoor sports.
5. Integrate technology and AI tools for health monitoring, performance analysis and wellness management.

UNIT 1	Health, Wellness and Preventive Healthcare
Topics	Concept of health, wellness and fitness – definitions and seven dimensions WHO definition of health; holistic health model (Physical, Mental, Social, Spiritual) Nutrition basics – macronutrients, micronutrients, balanced diet, BMI calculation Lifestyle diseases – diabetes, hypertension, obesity – causes and prevention Mental health & stress management – causes, symptoms, COPE model Sleep hygiene and circadian rhythm regulation Substance abuse – awareness, prevention and rehabilitation Personal hygiene, sanitation and communicable disease prevention Ergonomics and occupational health for students
Unit-wise Learning Outcomes (ULOs)	On completion of this unit, students will be able to: ULO1.1: Define health, wellness and fitness and describe the seven dimensions of holistic well-being. [Remember] ULO1.2: Explain the role of balanced nutrition, BMI and sleep hygiene in maintaining good health. [Understand] ULO1.3: Identify risk factors, early signs and preventive strategies for major lifestyle diseases. [Understand] ULO1.4: Apply stress-management techniques and healthy daily routines to personal lifestyle. [Apply] ULO1.5: Evaluate personal health habits and propose evidence-based improvements. [Evaluate]

UNIT 2	Yoga, Pranayama, Meditation and Mind-Body Practice
Topics	Introduction to Yoga – history, philosophy, Patanjali's Ashtanga Yoga Surya Namaskar – 12-step sequence with correct breathing

	Asanas for flexibility: Tadasana, Trikonasana, Paschimottanasana, Sarvangasana Asanas for strength & balance: Virabhadrasana I & II, Vrikshasana, Naukasana Pranayama techniques: Anulom-Vilom, Bhastrika, Kapalbhati, Bhramari Meditation and mindfulness – Dharana, Dhyana, body-scan, guided relaxation Yoga Nidra for sleep and stress relief Yoga for specific conditions – stress, back pain, PCOD/PCOS, anxiety Shatkarmas (cleansing processes) – an overview; International Day of Yoga
Unit-wise Learning Outcomes (ULOs)	On completion of this unit, students will be able to: ULO2.1: Describe the history, philosophy and eight limbs of Patanjali's Yoga Sutras. [Remember/Understand] ULO2.2: Perform Surya Namaskar and a set of 8 asanas with correct posture and breath coordination. [Apply] ULO2.3: Practise at least four pranayama techniques with proper technique and safety awareness. [Apply] ULO2.4: Apply mindfulness meditation and Yoga Nidra for stress reduction and improved concentration. [Apply] ULO2.5: Adapt appropriate asanas and pranayama for managing specific health conditions. [Analyse]

UNIT 3	Physical Fitness, Sports Skills and Sports Technology
Topics	Physical fitness components – strength, endurance, flexibility, agility, speed, coordination Principles of training – overload, progression, specificity, reversibility, individuality Warm-up, cool-down and dynamic stretching routines Indoor sports: Badminton, Table Tennis, Chess, Carrom – rules and basic skills Outdoor sports: Cricket, Football, Volleyball, Kabaddi, Athletics – rules and skills Sports injuries – types, RICE method, prevention and return-to-play protocol Sports psychology – motivation, goal setting, concentration and visualisation Gender equity and inclusivity in sports; Para-sports awareness Sports technology – wearables, fitness trackers, GPS analytics, performance dashboards
Unit-wise Learning Outcomes (ULOs)	On completion of this unit, students will be able to: ULO3.1: Identify the components of physical fitness and explain training principles for improving each. [Understand] ULO3.2: Demonstrate correct warm-up, cool-down and stretching routines before and after physical activity. [Apply] ULO3.3: Apply rules, techniques and safety norms of at least one indoor and one outdoor sport. [Apply] ULO3.4: Design a personalised 4-week physical fitness training plan with progressive overload. [Create] ULO3.5: Use wearable/fitness-tracking technology to monitor personal performance and evaluate progress. [Evaluate]

Course Outcomes (COs)

CO No.	Course Outcome	Bloom's Level
CO1	Explain the fundamental concepts of health, wellness, fitness and the dimensions of holistic well-being.	Remember/Understand
CO2	Demonstrate correct yogic practices – asanas, pranayama and meditation – for physical and mental health.	Apply
CO3	Design a personal fitness plan integrating nutrition, exercise, stress management and lifestyle modification.	Analyse/Apply
CO4	Understand rules, skills and safety norms for a chosen sport and participate actively with sportsmanship.	Apply
CO5	Evaluate personal health indicators and reflect on lifestyle choices using technology and evidence-based practices.	Evaluate

Bloom's Taxonomy Levels: Remember (L1) | Understand (L2) | Apply (L3) | Analyse (L4) | Evaluate (L5) | Create (L6)

CO – PO / PSO Mapping

Correlation: 3 = Strong | 2 = Moderate | 1 = Slight | – = No correlation

CO / PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
CO1	2	-	-	-	-	3	2	-	-	-	-	-	-	-
CO2	-	-	3	-	-	3	3	-	-	-	-	-	-	-
CO3	-	2	3	3	-	3	3	-	-	-	-	-	2	-
CO4	-	-	3	-	-	2	2	-	3	-	-	-	-	-
CO5	-	3	-	3	2	3	2	-	-	-	-	-	2	-

Text Books

S.No	Title	Author(s)	Publisher	Year
1	Health and Physical Education	Dr. S. K. Mangal & Shubhra Mangal	PHI Learning Pvt. Ltd.	2021
2	The Science of Yoga	William J. Broad	Simon & Schuster	2012

Reference Books

S.No	Title	Author(s)	Publisher	Year
1	Yoga: The Iyengar Way	Silva, Mira & Shyam Mehta	Dorling Kindersley	2019
2	Sports Medicine Essentials	Jim Clover	Cengage Learning	2020
3	Foundations of Physical Education & Sport	Charles A. Bucher	McGraw-Hill Education	2018
4	Mindfulness: An Eight-Week Plan for Finding Peace	Mark Williams & Danny Penman	Rodale Books	2011

Web Resources

1. Ministry of AYUSH – https://main.ayush.gov.in
2. Yoga for Youth (International Day of Yoga) – https://yoga.ayush.gov.in
3. National Health Portal – https://nhp.gov.in
4. ICMR Dietary Guidelines – https://www.icmr.nic.in
5. Sports Authority of India – https://www.sportsauthorityofindia.nic.in
6. WHO Physical Activity Guidelines – https://www.who.int/news-room/fact-sheets/detail/physical-activity
7. Khelo India Programme – https://kheloindia.gov.in
8. NIMHANS (Mental Health Resources) – https://nimhans.ac.in

YouTube Resources

1. Surya Namaskar – Complete 12-Step Tutorial – https://youtu.be/surya_tut_01
2. Pranayama for Beginners (Swami Ramdev) – https://youtu.be/pranayama_sr_02
3. Mindfulness Meditation (10 Min) – https://youtu.be/mindful_med_03
4. Balanced Nutrition Explained (ICMR) – https://youtu.be/nutrition_icmr_04
5. Sports Injury Prevention Tips – https://youtu.be/injury_prev_05
6. Mental Health Awareness for Students – https://youtu.be/mh_students_06
7. Yoga for Stress Relief – Isha Foundation – https://youtu.be/yoga_isha_07
8. Fit India Movement – PM Modi Launch – https://youtu.be/fitindia_pmo_08

AI-Integrated Activities

S.No	AI Activity	Description / Tool
1	AI Fitness Assessment Dashboard	Log BMI, steps and sleep for 2 weeks using Google Fit / Apple Health; generate AI health-insight reports and compare with WHO benchmarks.
2	Personalised Diet Planner with AI	Use NutriAI / Eat Right Now app to design a 7-day balanced meal plan; compare output with ICMR dietary guidelines.
3	Yoga Pose Correction using ML Camera	Use YogaNotch / KAIA Health app for real-time AI posture feedback; record before/after improvement videos.
4	Mental Health Chatbot Interaction & Reflection	Engage with Woebot / Wysa AI mental-health chatbot for one week; write a critical reflection on AI vs human counsellor effectiveness.
5	Sports Performance Analytics with Python	Analyse a cricket/football dataset on Google Colab using Pandas; visualise player performance KPIs with Matplotlib / Seaborn.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26HS0869) PROBABILITY AND STATISTICS
(Common to CSE & Allied)

Prerequisites: Set theory, Permutations and combinations, Integration

Course Objectives :

The objectives of this course is to

1. *Establish a firm foundation in probability theory, random variables, and their properties to model and quantify uncertainty in engineering contexts.*
2. *Analyse standard probability distributions and apply them to compute probabilities and expected values for real-world discrete and continuous phenomena.*
3. *Formulate and evaluate statistical inferences , including estimation, hypothesis testing, and correlation , to support data-driven engineering decisions.*

UNIT-I	**Descriptive Statistics**
Content	Introduction to statistics; population vs. sample; primary and secondary data; types of variables: categorical and continuous. Measures of central tendency: mean, median, mode; measures of variability: range, variance, standard deviation; skewness; kurtosis..
AI Integration	Python-based computation of mean, median, mode, variance, and standard deviation using NumPy and Pandas; visualization of data distributions with histograms and box plots using Matplotlib/Seaborn; AI-assisted interpretation of skewness and kurtosis in real datasets. .
Free/Open-Source AI Tools	Python (NumPy, Pandas, Matplotlib, Seaborn) on Google Colab; SciPy stats module

UNIT-II	**Probability and Random Variables**
Core Content	Sample space and events; axioms of probability; addition and multiplication theorems (without proof); conditional probability; Bayes' theorem (without proof).. Random variables: discrete and continuous; probability mass function; probability density function; cumulative distribution function; mathematical expectation and variance.
AI Integration	Python simulation of probability experiments using NumPy random module; visualization of PMF, PDF, and CDF using Matplotlib; AI-assisted derivation and verification of Bayes' theorem problems; simulation of conditional probability scenarios using SciPy.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy; WolframAlpha free tier.

UNIT-III	Probability Distributions
Core Content	Binomial distribution: properties mean and variance; Poisson distribution: properties, Poisson approximation to binomial. Normal distribution: properties, standard normal curve, areas under normal curve; uniform distribution.
AI Integration	Python-based generation and plotting of binomial, Poisson, normal, and uniform distributions using SciPy and Matplotlib; AI-assisted parameter estimation and verification; simulation of real-world problems using standard distributions. **Free/Open-Source AI Tools**:
Free/Open-Source AI Tools:	Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy; Scilab.

UNIT-IV	Sampling Theory and Estimation
Core Content	Population and samples; sampling distributions; central limit theorem: statement and significance; t-distribution; chi-square distribution; F-distribution. Point estimation: properties of estimators; interval estimation: confidence intervals for mean and proportion.
AI Integration	Python simulation of sampling distributions and central limit theorem convergence using NumPy; plotting t-, chi-square, and F-distributions with SciPy; AI-assisted construction and interpretation of confidence intervals..
Free/Open-Source AI Tools:	Python (NumPy, SciPy, Matplotlib, Pandas) on Google Colab; SymPy; GNU Octave.

UNIT-V	Testing of Hypothesis and Correlation
Core Content	Statistical hypothesis: null and alternative hypothesis; level of significance; large sample tests: z-test for means and proportions; small sample tests: t-test (one sample, two samples, paired); F-test. Chi-square test: goodness of fit, independence; correlation coefficient; rank correlation; regression lines and regression coefficients.
AI Integration	Python-based hypothesis testing using SciPy (z-test, t-test, F-test, chi-square test); AI-assisted interpretation of p-values and test statistics; computation and visualization of correlation matrices using Pandas and Seaborn; regression analysis using scikit-learn and Matplotlib.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Pandas, Matplotlib, Seaborn, scikit-learn); stats models library.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and describe measures of central tendency, dispersion, skewness, and kurtosis to summarise and characterise data sets.	**L1 (Remember)**
CO2	Explain the axioms and theorems of probability and the properties of discrete and continuous random variables including expectation and variance.	**L2 (Understand)**
CO3	Apply binomial, Poisson, normal, and uniform distributions to compute probabilities and derive distribution parameters for engineering problems.	**L3 (Apply)**
CO4	Analyse sampling distributions, central limit theorem implications, and construct confidence intervals for population parameters.	**L4 (Analyse)**
CO5	Evaluate statistical hypotheses using large and small sample tests and interpret correlation and regression relationships in data.	**L5 (Evaluate)**

Bloom's Taxonomy Coverage , Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1	Remember	**CO1**	Descriptive statistics measure identification and recall exercises.
L2	Understand	**CO2**	Probability axiom and random variable property explanation tasks.
L3	Apply	**CO3**	Distribution-based probability computation problems for binomial, Poisson, and normal cases.
L4	Analyse	**CO4**	Sampling distribution analysis; confidence interval construction exercises.
L5	Evaluate	**CO5**	Hypothesis test selection and evaluation; correlation/regression relationship interpretation.

CO-PO Articulation Matrix

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11
CO1	L1	3	1	,	,	,	,	,	,	,	,	,
CO2	L2	3	2	,	1	,	,	,	,	,	,	,
CO3	L3	3	2	1	1	1	,	,	,	,	,	,
CO4	L4	3	3	1	2	1	,	,	,	,	,	,
CO5	L5	3	3	2	3	1	1	,	,	1	1	,

CO-PO Weightage :

Average weightage based on the COs is shown below; it is aligned to NBA UG POs PO1 to PO11 and kept within the 1,3 correlation scale.

Avg	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
Avg	3.0	2.2	0.8	1.4	0.6	0.2	0	0	0.2	0.2	0

Textbooks :

1. Walpole, R. E., Myers, R. H., Myers, S. L., and Ye, K., *Probability & Statistics for Engineers and Scientists*, 9th Edition, Pearson, 2016.
2. Miller, I., and Freund, J. E., *Probability and Statistics for Engineers*, 8th Edition, Pearson, 2014.

Reference Books :

1. Jain, R. K., and Iyengar, S. R. K., *Advanced Engineering Mathematics*, 5th Edition, Alpha Science International, 2021.
2. Ross, S. M., *Introduction to Probability and Statistics for Engineers and Scientists*, 5th Edition, Academic Press, 2014.
3. Gubner, J. A., *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge University Press, 2006.
4. Johnson, R. A., and Bhattacharyya, G. K., *Statistics: Principles and Methods*, 7th Edition, Wiley, 2014.

Online Resources :

1. NPTEL: Probability and Statistics , https://nptel.ac.in
2. MIT OpenCourseWare , Introduction to Probability and Statistics: https://ocw.mit.edu
3. Python SciPy Statistics Documentation: https://docs.scipy.org/doc/scipy/reference/stats.html
4. Statsmodels Documentation: https://www.statsmodels.org
5. Google Colab for Python: https://colab.research.google.com

Web Resources : & NPTEL Links :

1. NPTEL: Probability and Statistics , https://nptel.ac.in
2. MIT OpenCourseWare , Introduction to Probability and Statistics: https://ocw.mit.edu
3. Python SciPy Statistics Documentation: https://docs.scipy.org/doc/scipy/reference/stats.html
4. Google Colab for Python: https://colab.research.google.com
5. NumPy Documentation: https://numpy.org/doc
6. Statsmodels Documentation: https://www.statsmodels.org
7. WolframAlpha (free tier) for verification: https://www.wolframalpha.com

Recommended Free & Open-Source AI/Computational Tools :

1. Python (NumPy, SciPy, Matplotlib, Pandas) , FREE on Google Colab for statistical computation, distribution analysis, and visualization.
2. SymPy (free, open-source) , For symbolic probability derivations, distribution functions, and algebraic verification.
3. Seaborn (free, open-source) , For statistical data visualization including correlation heatmaps, distribution plots, and regression plots.
4. scikit-learn (free, open-source) , For regression modelling, correlation analysis, and exploratory data analysis.
5. Jupyter Notebook (free, open-source) , For interactive statistical computation and lab documentation.
6. statsmodels (free, open-source) , For advanced hypothesis testing, regression analysis, and time series statistics.
7. GNU Octave (free, open-source) , MATLAB-like numerical computing for statistical and probability workflows..

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY::PUTTUR
(AUTONOMOUS)

II B.Tech - I Sem.

L	T	Pr	P	C
2	0	0	0	2

(26HS0811) BIOLOGY FOR ENGINEERS
(Common to All Branches)

Course Objectives

The objectives of this course are

1. Provide foundational biological knowledge with engineering relevance.
2. Demonstrate AI-driven applications in biology and bioengineering.
3. Encourage interdisciplinary teaching that connects biology, engineering, and computational tools.
4. Prepare students for innovation in healthcare, materials, and bio-industries.
5. Evaluate biomimetic principles and emerging bioengineering trends for real-world engineering applications.

Unit – I	**Foundations of Biology and AI**

Core Content	Cell as the basic unit of life. Carbohydrates, proteins, lipids, enzymes, vitamins, hormones — structure, function, and engineering relevance.
AI Integration	AI-based visualization of biomolecules, Case studies on AI in drug discovery.
Free/Open-Source AI Tools	AlphaFold (free via EBI); DeepChem (free, open-source); Python (NumPy, Matplotlib) on Google Colab (free); AI tools: ChatGPT / Gemini.

Unit – II	**Biomolecules in Engineering Applications**

Core Content	Carbohydrates in bioplastics (PHA, PLA). Proteins in food engineering (whey protein, plant analogs). Lipids in biodiesel and detergents. AI-driven generative design for biomaterials. **Nucleic Acids in Medicine- genetic vaccines.**
AI Integration	AI-driven generative design for biomaterials, Lipids in biodiesel & detergents, Genetic frameworks utilized in modern mRNA and DNA vaccines, as well as forensic DNA fingerprinting.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini.

Unit – III	Human Systems and Bio-designs

Core Content	Brain as CPU (EEG, robotic prosthetics). Eye as camera (optical corrections, lens materials). Heart as pump (ECG, stents). Kidney as filtration (dialysis systems). Endocrine System (Messaging)
AI Integration	AI-based ECG anomaly detection, Robotic arm design inspired by neural signals, Use of AI imaging tools to detect cataracts, simulate stent design using ANSYS, Similar to a network communication protocol.
Free/Open-Source AI Tools	Python (TensorFlow, scikit-learn, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini.

Unit – IV	Bio-inspired Materials and Mechanisms

Core Content	Photosynthesis → photovoltaic cells. Bird flight → GPS and aircraft navigation. Lotus leaf → superhydrophobic surfaces. Kingfisher beak → bullet train design.
AI Integration	Model solar energy conversion inspired by photosynthesis, AI simulation of lotus-leaf hydrophobicity, Kingfisher-inspired aerodynamic modelling, Use of AI to design self-cleaning surfaces.
Free/Open-Source AI Tools	Python (NumPy, SciPy, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini.

Unit – V	Emerging Trends in Bioengineering

Core Content	Muscular and skeletal scaffolds. 3D bioprinting of tissues (ear, bone, skin). AI in medical imaging (MRI, CT scans). AI in diagnostics and predictive healthcare.
AI Integration	AI in medical imaging (MRI, CT scans), AI in diagnostics and predictive healthcare.
Free/Open-Source AI Tools	Python (TensorFlow, OpenCV, Matplotlib) on Google Colab (free); ANSYS student version (free); AlphaFold (free via EBI); AI tools: ChatGPT / Gemini.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Explain the fundamentals of biology, biomolecules, and human systems relevant to engineering applications.	L2 (Understand)

CO2	Apply AI tools and computational techniques for biological data analysis, biomolecule visualisation, and healthcare applications.	L3 (Apply)
CO3	Analyse biomolecules and bio-inspired systems for solving engineering problems and developing sustainable technologies.	L4 (Analyse)
CO4	Design biomimetic and bioengineering solutions using interdisciplinary approaches and AI-enabled tools.	L5 (Evaluate) L6 (Create)
CO5	Evaluate emerging trends in bioengineering such as 3D bioprinting, regenerative medicine, and AI-based diagnostics for societal applications.	L5 (Evaluate)

CO – PO Articulation Matrix

Scale: 3 – String, 2 – Moderate, 1 – Slight

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	1	–	–	–	–	–	–	2	2	1
CO2	2	3	2	2	3	–	–	–	1	–	–	2	2	2
CO3	2	3	3	2	2	2	3	–	–	–	–	2	2	2
CO4	2	2	3	2	3	2	2	1	2	1	–	2	2	2
CO5	2	2	2	2	3	3	2	1	1	1	–	3	2	2

Text Books:

1. Singh R.C. and Rao N.R., Biology for Engineers, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.

2. Thyagarajan S. et al., Biology for Engineers, Tata McGraw-Hill, New Delhi, 2012.

References Books:

1.Fox S. and Rompolski K., Human Physiology, McGraw-Hill eBook, 16th Edition, 2022.

2. Bar-Cohen Y., Biomimetics: Nature-Based Innovation, 1st Edition, CRC Press, 2012.

3.Ozbolat I., 3D Bioprinting: Fundamentals, Principles and Applications, Academic Press, 2016.

Web References:

1. NPTEL Biology for Engineers – https://nptel.ac.in/courses/121106008

2. MIT OpenCourseWare – Introduction to Biological Engineering: https://ocw.mit.edu

3. AlphaFold Structure Database – https://alphafold.ebi.ac.uk

Other Resource (AI Tool, Open-Source Software etc.)

1. AlphaFold (free via EBI); DeepChem (free, open-source); Python (NumPy, Matplotlib) on Google Colab (free); AI tools: ChatGPT / Gemini

2. Python (NumPy, SciPy, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini

3. Python (TensorFlow, scikit-learn, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini

4. Python (NumPy, SciPy, Matplotlib) on Google Colab (free); ANSYS student version (free); AI tools: ChatGPT / Gemini

5. Python (TensorFlow, OpenCV, Matplotlib) on Google Colab (free); ANSYS student version (free); AlphaFold (free via EBI); AI tools: ChatGPT / Gemini.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0505) DIGITAL LOGIC AND COMPUTER ORGANIZATION
(CSE & Allied Branches)

Prerequisites: Computational Thinking and Problem Solving with Python

Course Objectives:

The objectives of the course is to

1. *Understand number systems, binary codes, Boolean algebra, and logic gates; apply K-map minimization and universal NAND/NOR gates to simplify and implement digital logic functions.*
2. *Analyse and design combinational circuits and sequential circuits.*
3. *Understand Von-Neumann architecture, functional units, and bus structures; apply computer arithmetic techniques.*
4. *Understand memory organization covering RAM, ROM, cache mapping, virtual memory, and secondary storage; implement programmable logic using PLA, PAL, and FPGA.*
5. *Understand I/O organization including programmed I/O, interrupt-driven I/O, DMA transfer modes, bus structures, interface circuits, and standard interfaces (USB and PCI).*

UNIT-I	**Data Representation and Digital Logic Circuits , I**
Content	Binary Numbers, Fixed Point Representation, Floating Point Representation, Number base conversions, Octal and Hexadecimal Numbers, Signed binary numbers, Binary codes. Basic Logic Functions, Logic gates, Universal logic gates, Minimization of Logic expressions, K-Map Simplification, Comparator, Decoders, Multiplexers.
AI Integration	Use an AI/LLM tool to explain binary, octal, decimal, and hexadecimal number systems and walk through conversion steps; solve problems on fixed-point/floating-point representation and complement arithmetic. Use a Python template in Google Colab to verify K-map simplification against manually derived truth tables. Use an AI/LLM tool to design a 2-to-4 decoder and 4-to-1 multiplexer from NAND gates and trace circuit outputs.

UNIT-II	**Digital Logic Circuits , II and Basic Structure of Computers**
Content	Sequential Circuits, Flip-Flops (SR, JK, D, T), master-slave flip-flops, flip-flop conversions, Binary counters, Registers, Shift Registers (SISO, SIPO, PISO, PIPO), Ripple counters, Synchronous counters. Computer Types, Functional units, Basic operational concepts, Bus

	structures, Software, Performance, Multiprocessors and multicomputers, Computer Generations, Von-Neumann Architecture.
AI Integration	Use an AI/LLM tool to explain flip-flop truth tables, excitation tables, and conversion methods (e.g. JK to D). Use a Python template in Google Colab to simulate a 4-bit synchronous up-counter and SIPO shift register, verifying state sequences. Use an AI/LLM tool to explain Von-Neumann architecture, fetch-decode-execute cycle, and compare with Harvard architecture.

UNIT-III	**Computer Arithmetic**
Content	Addition and Subtraction of Signed Numbers, Design of Fast Adders (Carry-Lookahead Adder), Multiplication of Positive Numbers, Signed-operand Multiplication (Booth's Algorithm). Fast Multiplication (Wallace Tree), Integer Division (Restoring and Non-Restoring), Floating-Point Numbers and Operations (IEEE 754).
AI Integration	Use an AI/LLM tool to explain 2's complement addition/subtraction with overflow detection and walk through Carry-Lookahead Adder design, comparing delay versus ripple carry adders. Use a Python template in Google Colab to perform Booth's algorithm multiplication and compare with shift-and-add method. Use an AI/LLM tool to explain IEEE 754 floating-point format with a worked addition example.

UNIT-IV	**Memory Organization and Programmable Logic**
Content	Basic Concepts of Memory Organization, Semiconductor RAM Memories (SRAM and DRAM), Read-Only Memories (ROM, PROM, EPROM, EEPROM, Flash), Speed, Size and Cost. Cache Memories (direct-mapped, set-associative, fully associative), Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage; Programmable Logic Array (PLA), Programmable Array Logic (PAL), FPGA.
AI Integration	Use an AI/LLM tool to explain memory hierarchy and cache operation (hit, miss, LRU/FIFO) with a direct-mapped cache example, calculating hit rate manually. Use a Python template to simulate direct-mapped vs 2-way set-associative cache for a memory address trace. Use an AI/LLM tool to implement a Boolean function using PLA and PAL programming tables and compare flexibility.

UNIT-V	Input/Output Organization and Interfacing
Content	Introduction to Input/Output organization, Accessing I/O devices, Programmed I/O, Interrupt-driven I/O, Interrupt handling, Vectored interrupts, Processor examples for I/O operations. Direct Memory Access (DMA), DMA transfer modes (burst, cycle-steal, transparent), Buses and bus structures, Synchronous and asynchronous buses, Interface circuits, Serial and parallel communication, Standard I/O interfaces, USB, PCI, Memory-mapped and isolated I/O.
AI Integration	Use an AI/LLM tool to compare programmed I/O, interrupt-driven I/O, and DMA in terms of CPU involvement and throughput, and trace the complete interrupt handling sequence. Use a Python template to simulate programmed vs interrupt-driven I/O and compute CPU utilization for both. Use an AI/LLM tool to explain synchronous/asynchronous bus timing and compare USB/PCI interface standards.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply number systems, binary codes, Boolean algebra, and logic gate principles to represent, convert, and simplify digital logic functions using K-maps.	**L3 (Apply)**
CO2	Design combinational circuits , adders, comparators, multiplexers, and decoders , and analyze sequential circuits including flip-flops, counters, and shift registers.	**L4 (Analyze)**
CO3	Apply computer arithmetic techniques , 2's complement arithmetic, carry-lookahead addition, Booth's multiplication, and IEEE 754 floating-point operations.	**L3 (Apply)**
CO4	Explain memory organization including semiconductor memories, cache memory, virtual memory, and programmable logic devices (PLA, PAL, FPGA).	**L2,L3**
CO5	Describe I/O organization methods , programmed I/O, interrupt-driven I/O, and DMA , and explain bus structures, interface circuits, and standard I/O interfaces (USB, PCI).	**L2,L3**

Bloom's Taxonomy Coverage , Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L2/L3	Understand/Apply	**CO4, CO5**	Memory hierarchy and cache mapping explanation; I/O method comparison and bus structure description.

L3	Apply	**CO1, CO3**	K-map simplification problems; Booth's algorithm and IEEE 754 floating-point exercises.
L4	Analyze	**CO2**	Sequential circuit design and flip-flop conversion analysis; counter and shift register behaviour evaluation.

CO , PO Articulation Matrix (NBA)

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**2**	1	1	**2**	,	,	,	1	1	,	1
CO2	**3**	**3**	**2**	1	**2**	,	,	,	1	1	,	**2**
CO3	**3**	**3**	**2**	1	**2**	,	,	,	1	1	,	1
CO4	**3**	**2**	1	1	**2**	,	,	,	1	**2**	,	**2**
CO5	**3**	**2**	1	1	**2**	,	,	,	1	**2**	,	**2**

CO-PO Correlation Highlights	CO1 and CO2 strongly align with PO1 and PO2 through logic minimization and sequential circuit design. CO3 maps to PO2 and PO5 through arithmetic algorithm implementation and AI-assisted simulation. CO4 and CO5 connect to PO1 and PO5 through memory and I/O system explanation supported by AI/LLM-guided conceptual tasks and Python simulation across all five units. PO9 (Teamwork) and PO10/PO12 are addressed through collaborative lab simulation exercises and continued engagement with evolving computer architecture concepts.

Textbooks :

1. Hamacher, C., Vranesic, Z., and Zaky, S., *Computer Organization*, 6th Edition, McGraw Hill, 2023.
2. Mano, M. M., *Digital Design*, 6th Edition, Pearson Education, 2018.

Reference Books :

1. Mano, M. M., *Computer Systems Architecture*, 3rd Edition, Pearson, 2017.
2. Patterson, D. A., and Hennessy, J. L., *Computer Organization and Design*, Elsevier, 2004.
3. Roth, C. H., *Fundamentals of Logic Design*, 5th Edition, Thomson, 2003.
4. Stallings, W., *Computer Organization and Architecture*, 11th Edition, Pearson, 2022.

Online Resources :

1. NPTEL Computer Organization: https://nptel.ac.in
2. NPTEL Digital Logic Design: https://nptel.ac.in
3. Google Colab for Python: https://colab.research.google.com
4. VisuAlgo , Visualizing Data Structures: https://visualgo.net

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech - I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0506) ALGORITHMS DESIGN AND ANALYSIS
(Professional Core-II)
(CSE & Allied Branches)

Course Objectives:

The objectives of the course is to

1. *Understand fundamental algorithm design paradigms , divide and conquer, greedy, dynamic programming.*
2. *Analyse algorithm efficiency and correctness using asymptotic notation, recurrence relations, and amortized analysis.*
3. *Design efficient algorithms for complex problems using appropriate paradigms.*
4. *Understand computational complexity, NP-completeness, and approximation strategies.*

UNIT-I	ALGORITHM ANALYSIS AND MATHEMATICAL FOUNDATIONS
Content	**Asymptotic notation:** O, Ω, Θ, o, ω; rate of growth comparisons; properties of asymptotic notations; amortized analysis , aggregate, accounting, and potential methods. Recurrence solving: substitution method; recursion tree method; Master theorem , all three cases, extensions, and limitations. Lower bounds: decision tree model; lower bounds for sorting (Ω(n log n)) and searching; adversary arguments. **Source:** TB1: Chapters 1,4 (Cormen et al., CLRS 4th ed.); TB2: Chapter 5 (Kleinberg & Tardos)
AI Integration	Use VisuAlgo and Python Tutor to visualize recursion trees for Merge Sort and Binary Search; use Wolfram Alpha and SymPy to solve recurrence equations; plot and compare O(log n), O(n), O(n log n), O(n^2) growth curves.

UNIT-II	DIVIDE AND CONQUER
Content	**Divide and conquer paradigm:** strategy, recurrence formulation, subproblem independence. Sorting: Merge Sort (algorithm, analysis, stable), Quick Sort (partitioning schemes, randomized Quick Sort, expected complexity), Heap Sort review. Searching and selection: Binary Search and variants; finding median and order statistics , randomized selection (expected O(n)); deterministic O(n)

	selection. **Advanced applications:** Strassen's matrix multiplication , recurrence, complexity, cache performance; convex hull introduction; Master theorem applications for all divide-and-conquer recurrences. **Source:** TB1: Chapters 7,9, 28 (Cormen et al.); TB2: Chapter 5 (Kleinberg & Tardos)
AI Integration	Use VisuAlgo and Algorithm Visualizer to animate Merge Sort, Quick Sort, and Heap Sort; compare Quick Sort pivot strategies (first, last, random median); trace Strassen matrix multiplication and compare with classical $O(n^3)$.

UNIT-III	**GREEDY ALGORITHMS**
Content	Greedy method fundamentals: greedy choice property, optimal substructure, control abstraction. Classic problems: activity selection, fractional knapsack, job sequencing with deadlines , greedy proof techniques. Graph algorithms: Minimum Spanning Tree , Prim's algorithm (priority queue implementation, O(E log V)), Kruskal's algorithm (disjoint set union-find); shortest path , Dijkstra's algorithm (non-negative weights); correctness proofs. Advanced: biconnected components; Huffman coding , optimal prefix-free codes, greedy construction, proof of optimality. **Source:** TB1: Chapters 16, 23,24 (Cormen et al.); TB2: Chapters 4, 6 (Kleinberg & Tardos)
AI Integration	Use Gephi and Cytoscape to visualize MST and shortest path algorithms; use VisuAlgo to animate Prim's, Kruskal's, and Dijkstra's; compare multiple scheduling strategies in activity selection.

UNIT-IV	**DYNAMIC PROGRAMMING**
Content	**DP principles:** optimal substructure, overlapping subproblems; memoization (top-down) vs. tabulation (bottom-up); reconstructing solutions from DP tables. **Classic DP problems:** Longest Common Subsequence (LCS) , table construction and traceback; 0/1 Knapsack , DP table filling; Edit Distance (Levenshtein) , string alignment; Optimal Binary Search Tree. **Advanced DP:** Floyd-Warshall all-pairs shortest path , $O(V^3)$, negative edge handling, detecting negative cycles; Travelling Salesman Problem (TSP) , Held-Karp DP (exact); Subset Sum. **Source:** TB1: Chapters 15, 25 (Cormen et al.); TB2: Chapter 6 (Kleinberg & Tardos)

AI Integration	Use DP Visualizer and Python Tutor to animate LCS table and knapsack DP matrix; compare memoization vs. tabulation on execution time and memory; use ChatGPT to verify step-by-step DP trace for edit distance.

UNIT-V	**ADVANCED TOPICS AND COMPUTATIONAL COMPLEXITY**
Content	**Backtracking:** N-Queens problem, graph coloring, Hamiltonian cycle , pruning and state space trees. Branch and Bound: TSP , bounding functions, LC-search; comparison with backtracking. **Computational complexity:** problem classes P, NP, NP-Complete, NP-Hard; polynomial reduction; Cook's theorem (SAT is NP-Complete); classic NP-complete problems , Vertex Cover, Clique, 3-SAT, Hamiltonian Path. **Approximation algorithms:** 2-approximation for Vertex Cover; greedy set cover; TSP approximation (Christofides-Serdyuk); randomized algorithms: Las Vegas vs. Monte Carlo; Randomized QuickSort expected analysis. **Source:** TB1: Chapters 34,35 (Cormen et al.); TB2: Chapter 8 (Kleinberg & Tardos)
AI Integration	Use Graphviz and Gephi to visualize backtracking state space trees; animate N-Queens and graph coloring; classify 10 problems into P/NP/NP-Hard/NP-Complete using Claude; compare exact vs. approximation TSP solutions.

Course Outcomes (COs

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

CO	**Course Outcome Statement**	**Bloom's Taxonomy Level**
CO1	Apply asymptotic notation, amortized analysis, and recurrence solving techniques to analyse algorithm complexity. (L3)	L4 (Analyse)
CO2	Design and analyse divide-and-conquer algorithms , sorting, selection, and matrix multiplication. (L4)	L4 (Analyse)
CO3	Apply greedy algorithms to graph and optimization problems , MST, shortest path, and scheduling. (L3)	L4 (Analyse)
CO4	Design dynamic programming solutions for classic problems , LCS, knapsack, edit distance, and shortest paths. (L4)	L5 (Evaluate)
CO5	Classify problems into P, NP, NP-Complete, NP-Hard; apply backtracking, branch-and-bound, and approximation algorithms. (L4)	L5 (Evaluate)

Bloom's Taxonomy Coverage , Summary

Bloom's Level	Level Name	CO(s)	Sample Activities Assessments
L1,L2	Remember Understand	**CO1**	Define asymptotic notations; describe divide-and-conquer, greedy, DP paradigms.
L3	Apply	**CO1, CO2**	Implement binary search, merge sort, quick sort, and Strassen's algorithm.
L4	Analyse	**CO1,CO3**	Analyse recurrence relations; compare Greedy vs. DP approaches.
L5	Evaluate	**CO4, CO5**	Evaluate algorithm selection for a given problem; assess NP-completeness.
L6	Create	**CO3, CO5**	Design a novel algorithm for a real-world optimisation problem.

CO , PO PSO Articulation Matrix (NBA)

Scale: 3 , Strong | 2 , Moderate | 1 , Slight | , No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO 1	L4	**3**	**3**	**2**	**3**	**2**	,	,	,	,	1	,
CO 2	L4	**3**	**3**	**2**	**3**	**2**	,	,	,	,	1	,
CO 3	L4	**3**	**3**	**3**	**3**	**2**	,	,	,	1	**2**	,
CO 4	L5	**3**	**3**	**3**	**3**	**3**	,	,	,	1	**2**	1
CO 5	L5	**3**	**3**	**3**	**3**	**3**	1	,	,	1	**2**	1

CO-PO Correlation Highlights	Strong PO2 (Problem Analysis) and PO4 (Investigation) through complexity analysis, recurrence solving, and empirical benchmarking. PO3 (Design/Development) through algorithm design for CO3 and CO5. PO5 (Modern Tools) through VisuAlgo, Python Tutor, Gephi, and Graphviz across all units.

Textbooks :

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, 4th Edition, MIT Press, 2022.
2. Kleinberg, J., and Tardos, E., *Algorithm Design*, Pearson Education, 2013.

Reference Books :

1. Dasgupta, S., Papadimitriou, C., and Vazirani, U., *Algorithms*, McGraw-Hill, 2008.
2. Sedgewick, R., and Wayne, K., *Algorithms*, 4th Edition, Addison-Wesley, 2011

Online Resources :

1. NPTEL , Design and Analysis of Algorithms, IIT Madras , nptel.ac.in
2. MIT OpenCourseWare , Introduction to Algorithms (6.006) , ocw.mit.edu
3. VisuAlgo , visualgo.net | Algorithm Visualizer , algorithm-visualizer.org
4. Coursera , Algorithms Specialization, Stanford coursera.org/specializations/algorithms

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0507) OBJECT ORIENTED DESIGN AND PROGRAMMING
(Professional Core -III)
(Common to CSE & Allied Branches)

Course Objectives

The objectives of the course is to

1. *Identify Java language components and understand how they work together in applications.*
2. *Learn the fundamentals of object-oriented programming , defining classes, invoking methods, using class libraries.*
3. *Extend Java classes with inheritance, dynamic binding, and exception handling.*
4. *Design applications with threads and understand Java concurrency mechanisms.*
5. *Apply Java APIs for file handling, GUI programming with JavaFX, and database connectivity.*

UNIT-I	OOP CONCEPTS, DATA TYPES, OPERATORS, AND CONTROL STATEMENTS
Content	**OOP principles:** encapsulation, inheritance, polymorphism, abstraction; Java program structure , tokens, statements, command-line arguments, escape sequences, comments, programming style. **Data types, variables, operators:** primitive types, type casting, scope, literals, symbolic constants, static variables, printf(); operators , arithmetic, increment/decrement, ternary, relational, logical, bitwise; precedence and associativity. **Control statements:** if, if-else, nested if, switch; iteration , while, do-while, for, for-each, nested loops; break, continue. **Source:** TB1: Chapters 1,4 (Seth & Juneja, JAVA one step ahead); TB2: Chapters 1,3 (Samanta & Sarma)
AI Integration	Use ChatGPT or GitHub Copilot to generate Java programs for control statements; use draw.io to design flowcharts for conditional and loop constructs; use AI to generate test cases for operators and validate precedence via execution.

UNIT-II	CLASSES, OBJECTS, AND METHODS
Content	**Classes and objects:** class declaration, modifiers, class members, object creation, assigning objects; access control (public, private, protected, default); constructors, overloaded constructors; nested classes, final class; passing by value and reference; keyword this. **Methods:** defining methods, overloaded methods, class objects as parameters, access control, recursive methods, nesting of methods, overriding methods; attributes final and static. Java Collections framework: ArrayList, LinkedList, HashMap, HashSet, TreeMap , selection based on use case. **Source:** TB1: Chapters 5,7 (Seth & Juneja); TB3: Chapters 8,10 (Deitel & Deitel, Java 9 for Programmers)
AI Integration	Use StarUML to design UML class diagrams for Student Management and Banking systems; use ChatGPT to generate class structures (constructors, methods) and refine manually; use draw.io for object interaction diagrams.

UNIT-III	ARRAYS, INHERITANCE, AND INTERFACES
Content	**Arrays:** declaration, initialization, storage in memory, element access, operations, sorting (Arrays.sort), searching; 2D and 3D arrays; arrays as vectors. **Inheritance:** process, types (single, multilevel, hierarchical), super class Object, inhibiting with final, access control, keyword super, constructor chaining, method overriding, dynamic method dispatch; abstract classes. **Interfaces:** declaration, implementation, multiple interfaces, nested interfaces, interface inheritance; default and static methods in interfaces; functional interfaces; annotations. **Source:** TB1: Chapters 8,11 (Seth & Juneja); TB2: Chapters 4,6 (Samanta & Sarma)
AI Integration	Use draw.io to create UML inheritance hierarchy diagrams (Vehicle → Car → ElectricCar); use AI to generate interface-based programs and modify for multiple inheritance; use ChatGPT to compare abstract class vs. interface with concrete examples.

UNIT-IV	PACKAGES, EXCEPTION HANDLING, AND JAVA I/O
Content	**Packages:** defining and importing packages, class path, access control; Java.lang , Object class, Enumeration, Math, Wrapper classes, auto-boxing/unboxing; java.util , Formatter, Random, Time package (Instant, TemporalAdjusters); date/time formatting. **Exception handling:** hierarchy of exception classes; keywords throws, throw, try, catch, finally; multiple catch clauses; class Throwable; checked

	vs. unchecked exceptions; try-with-resources. **Java I/O:** standard I/O streams; byte streams, character streams; Scanner class; reading and writing files using Java NIO (Files class). **Source:** TB1: Chapters 12,14 (Seth & Juneja); TB3: Chapter 15 (Deitel & Deitel) , Java I/O and Files
AI Integration	Use draw.io to design multi-package system architecture; use ChatGPT to generate programs with exceptions, identify errors, and fix them; use GitHub Copilot for file I/O programs; use AI to explore Java API (Math, Random, Time).

UNIT-V	**STRINGS, MULTITHREADING, JDBC, AND JAVAFX**
Content	**String handling:** CharSequence interface, String class, StringBuffer; methods , extraction, comparison, modification, searching. **Multithreaded programming:** Thread class, creating threads, thread states, priority, synchronization (synchronized keyword), deadlock and race conditions, inter-thread communication (wait, notify, notifyAll); suspending, resuming, stopping threads. **JDBC:** architecture, MySQL connector setup, establishing connections, Statement vs. PreparedStatement, ResultSet; CRUD operations. JavaFX GUI: Scene Builder, App window structure, displaying text and images, event handling, node layout, mouse events. **Source:** TB3: Chapters 11,12, 20 (Deitel & Deitel) , Strings, Multithreading, JDBC, JavaFX
AI Integration	Use ChatGPT to generate and debug multithreaded programs; simulate deadlock and resolve; build a JDBC mini-project , connect Java to MySQL and perform CRUD; design a JavaFX login form using Scene Builder.

Course Outcomes (Cos) :

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

CO	**Course Outcome Statement**	**Bloom's Taxonomy Level**
CO1	Analyse problems, design solutions using OOP principles, and implement them in Java. (L4)	L4 (Analyse)
CO2	Design and implement classes to model real-world entities with constructors, methods, and access control. (L4)	L4 (Analyse)
CO3	Demonstrate inheritance hierarchies, polymorphic behaviour, method overriding, and dynamic dispatch. (L3)	L3 (Apply)
CO4	Apply exception handling to write robust, fault-tolerant Java programs. (L3)	L3 (Apply)
CO5	Perform Java I/O, JDBC database connectivity, and build GUI	L3 (Apply)

	applications using JavaFX. (L3)	
CO6	Select appropriate Java collection data structures to solve computational problems. (L5)	L4 (Analyse)

CO , PO PSO Articulation Matrix (NBA)

Scale: 3 , Strong | 2 , Moderate | 1 , Slight | , No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO 1	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	**2**	,
CO 2	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	1	,
CO 3	L3	**3**	**3**	**2**	1	**2**	,	,	,	,	1	,
CO 4	L3	**3**	**3**	**2**	1	**2**	,	,	,	,	1	,
CO 5	L3	**3**	**2**	**2**	1	**2**	,	,	,	,	1	,
CO 6	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	1	,

CO-PO Correlation Highlights	Strong PO1 and PO3 (Design/Development) through OOP class design, inheritance, and JavaFX application development. PO2 (Problem Analysis) through analysing static vs. dynamic dispatch and collection performance. PO5 (Modern Tools) through StarUML, GitHub Copilot, IntelliJ IDEA, Scene Builder, and JDBC.

Textbooks :

1. Seth, A., and Juneja, B. L., *JAVA One Step Ahead*, Oxford University Press.
2. Samanta, D., and Sarma, M., *Joy with JAVA: Fundamentals of Object Oriented Programming*, Cambridge University Press, 2023.
3. Deitel, P., and Deitel, H., *Java 9 for Programmers*, 4th Edition, Pearson Education.

Reference Books :

1. Schildt, H., *The Complete Reference Java*, 11th Edition, McGraw-Hill.
2. Liang, Y. D., *Introduction to Java Programming*, 7th Edition, Pearson Education.

Online Resources :

1. https://www.youtube.com/@JoyofLearningCSE/featured
2. NPTEL , Programming in Java , nptel.ac.in/courses/106/105/106105191/
3. Oracle Java Documentation , docs.oracle.com/en/java/
4. StarUML , staruml.io | draw.io , draw.io | GitHub Copilot , github.com/features/copilot

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
0	0	0	2	1

(26CS0508) DIGITAL LOGIC AND COMPUTER ORGANIZATION LAB
(CSE & ALLIED BRANCHES)

Prerequisite: Digital Logic and Computer Organization (co-requisite)

List of Experiments

Unit 1	Number Systems, Logic Gates and K-Map Simplification

1. Number System Conversion and Boolean Expression Verification

Use a Python template in Google Colab to input a decimal number and observe its binary, octal, and hexadecimal equivalents; verify conversions manually using base conversion algorithms. Input a 4-variable Boolean expression and generate its truth table; compare with manually derived truth table.

2. K-Map Simplification and Logic Gate Verification

Simulate a 4-variable K-map for a given Boolean function, observe groupings, and derive the minimized SOP expression; verify by comparing with the manual K-map solution. Implement the simplified expression using basic logic gates (AND, OR, NOT) on a simulator and verify output for all input combinations.

3. Universal Gate Implementation , Decoder and Multiplexer Design

Design a 2-to-4 decoder and a 4-to-1 multiplexer using only NAND gates. Trace through the circuit for a given input combination and verify the output against the expected truth table; document the universal gate equivalences used (NOT, AND, OR from NAND).

Unit 2	Sequential Circuits and Computer Architecture Simulation

1. Flip-Flop Truth Table Verification and Conversion

Implement SR, JK, D, and T flip-flops on a digital logic simulator; verify their truth tables and excitation tables. Perform a flip-flop conversion exercise (e.g., JK to D) using the excitation table method, and verify the converted flip-flop's behaviour against the target flip-flop's truth table.

2. Synchronous Counter and Shift Register Simulation

Use a Python template in Google Colab to simulate a 4-bit synchronous up-counter and a 4-bit SIPO shift register; apply clock pulses one at a time and observe the state sequence. Verify the counter follows the correct binary sequence and that the shift register correctly loads serial input into parallel output.

3. Von-Neumann Architecture and Fetch-Decode-Execute Cycle Simulation

Trace through a simplified fetch-decode-execute cycle simulation for a sample instruction set, identifying the role of each functional unit (ALU, Control Unit, Memory, I/O).

Compare Von-Neumann and Harvard architecture data flow paths and document differences in a summary table.

Unit 3	Computer Arithmetic, Memory, and I/O Simulation

1. Carry-Lookahead Adder Design and Performance Comparison

Design a 4-bit Carry-Lookahead Adder (CLA) by deriving Generate and Propagate expressions and computing all carry signals in parallel. Compare the propagation delay of the CLA implementation against a ripple carry adder for the same 4-bit addition example.

2. Booth's Algorithm Multiplication Simulation

Using a Python template in Google Colab, perform binary multiplication of two 4-bit signed numbers using both the shift-and-add method and Booth's algorithm. Observe the partial products at each step and compare the number of addition steps required by both methods.

3. IEEE 754 Floating-Point Representation and Addition

Convert given decimal numbers to IEEE 754 single-precision floating-point format (sign, exponent, mantissa); perform a floating-point addition example step by step (align exponents, add mantissas, normalize, round). Verify special cases: zero, infinity, and NaN representation.

4. Cache Memory Mapping Simulation

Using a Python template, simulate a direct-mapped cache for a given set of memory addresses; observe which addresses cause hits and which cause misses, and compute the hit rate. Repeat for a 2-way set-associative cache with the same address trace and compare hit rates between the two mapping techniques.

5. Programmed I/O vs Interrupt-Driven I/O Simulation

Using a Python template, simulate programmed I/O (polling) and interrupt-driven I/O for a simple data transfer scenario. Observe and compute the number of CPU cycles consumed by each method for the same data size, and explain why DMA is preferred for high-speed bulk transfers.

AI Tools Integration (Lab)

AI Tools Integration (Lab)	Use AI/LLM tools (ChatGPT, Claude) for explaining number conversions, K-map groupings, flip-flop conversions, and IEEE 754 representation step by step. Python (Google Colab) for simulating counters, shift registers, cache mapping, and I/O transfer methods. Logic circuit simulators (Logisim, CircuitVerse) for combinational and sequential circuit design and verification.

Lab Course Outcomes

After successful completion of the lab, the student will be able to:

CO	Course Outcome Statement	Bloom's Level
CO1	Design and simplify combinational logic circuits using Boolean algebra, K-maps, and universal gates.	**L3 (Apply)**
CO2	Implement and verify the behaviour of sequential circuits including flip-flops, counters, and shift registers.	**L3,L4**
CO3	Simulate computer arithmetic operations including signed addition, multiplication, and IEEE 754 floating-point representation.	**L3,L4**
CO4	Simulate cache memory mapping techniques and compare their performance for given address traces.	**L4 (Analyze)**
CO5	Simulate I/O transfer methods (programmed, interrupt-driven, DMA) and compare CPU utilization across methods.	**L4 (Analyze)**

CO , PO Articulation Matrix (NBA)

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**2**	1	,	**2**	,	,	,	,	,	,	
CO2	**3**	**3**	**2**	1	**2**	,	,	,	1	,	,	
CO3	**3**	**3**	**2**	1	**2**	,	,	,	,	,	,	
CO4	**2**	**3**	1	**2**	**2**	,	,	,	,	,	,	
CO5	**2**	**3**	1	**2**	**2**	,	,	,	,	,	,	

CO-PO Correlation Highlights	CO1 and CO2 align strongly with PO1 and PO2 through logic design and sequential circuit verification. CO3 connects to PO2 and PO5 through arithmetic algorithm simulation. CO4 and CO5 strongly map to PO4 (Investigation) and PO5 (Modern Tools) through comparative performance analysis of cache mapping and I/O methods using Python-based simulation.

Textbooks :

1. Hamacher, C., Vranesic, Z., and Zaky, S., *Computer Organization*, 6th Edition, McGraw Hill, 2023.
2. Mano, M. M., *Digital Design*, 6th Edition, Pearson Education, 2018.

Reference Books :

1. Mano, M. M., *Computer Systems Architecture*, 3rd Edition, Pearson, 2017.
2. Patterson, D. A., and Hennessy, J. L., *Computer Organization and Design*, Elsevier, 2004.

Online Resources :

1. CircuitVerse , Online Digital Logic Simulator: https://circuitverse.org
2. Logisim Evolution , Open Source Logic Simulator
3. NPTEL Digital Logic Design: https://nptel.ac.in
4. Google Colab for Python: https://colab.research.google.com

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0509) ALGORITHMS DESIGN AND ANALYSIS LAB
(Professional Core -II Lab)
(Common to CSE & Allied Branches)

Course Objectives:

The objectives of the course is to

1. *Understand algorithm design paradigms including divide and conquer, greedy, dynamic programming, and approximation methods.*
2. *Develop the ability to analyse algorithm correctness and efficiency using asymptotic notation, recurrence relations, and amortized analysis.*
3. *Design efficient algorithmic solutions for optimization and computational problems.*
4. *Apply advanced algorithmic strategies including graph algorithms, backtracking, and branch-and-bound.*
5. *Utilize AI-assisted tools for visualization, implementation, complexity analysis, and performance evaluation of algorithms.*

List of Experiments

MODULE I: ALGORITHM ANALYSIS AND MATHEMATICAL FOUNDATIONS

Topics: Asymptotic growth rates, recurrence relations, recursion tracing, and self-balancing tree structures, building the mathematical foundation needed for algorithm analysis.

AI Integration: AI tools are used to visualize complexity growth trends, verify asymptotic notation, derive and validate recurrence solutions against the Master theorem, generate recursion trees, and visualize AVL/B-Tree balancing operations.

Suggested Tools: VisuAlgo; Python Tutor; Wolfram Alpha; SymPy; ChatGPT

Experiments:

- **Experiment-1: Complexity Growth Comparison** Implement algorithms representing O(log n), O(n), O(n log n), and $O(n^2)$ growth, execute them for varying input sizes, plot execution time, and compare practical versus theoretical growth using VisuAlgo and Python Tutor.
- **Experiment-2: Recurrence Relation Verification** Solve recurrence equations such as $T(n)=T(n-1)+1$, $T(n)=2T(n/2)+n$, and $T(n)=T(n/2)+1$, and verify the solutions experimentally, using Wolfram Alpha, SymPy, and ChatGPT to generate derivations and validate Master theorem solutions.
- **Experiment-3: Recursion Tree Visualization** Implement Merge Sort and Binary Search, trace their recursive execution, compare runtime, and generate recursion trees using VisuAlgo and Python Tutor.

- **Experiment-4: AVL Tree and B Tree Analysis** Implement AVL tree insertion with rotations and B Tree insertion, compare balancing operations, and use VisuAlgo and ChatGPT to generate insertion sequences and visualize balancing.

MODULE II: DIVIDE AND CONQUER

Topics: Divide-and-conquer strategy applied to sorting, order-statistics selection, and matrix multiplication, with emphasis on comparing recursive decomposition and measured performance.

AI Integration: AI tools help compare pivot strategies, generate test datasets and edge cases, compare expected versus observed complexity, and trace the recursive decomposition used in matrix multiplication.

Suggested Tools: VisuAlgo; Algorithm Visualizer; ChatGPT

Experiments:

- **Experiment-5: Sorting Algorithms Comparison** Implement Merge Sort, Quick Sort, and Heap Sort, and compare their runtime, number of comparisons, and swaps, using AI tools to compare pivot strategies and generate datasets.
- **Experiment-6: Selection Algorithms** Implement Randomized Selection and Deterministic Selection to find the median and kth smallest element, using AI tools to generate edge cases and compare expected complexity.
- **Experiment-7: Matrix Multiplication Optimization** Implement Classical Matrix Multiplication and Strassen's Algorithm, compare execution time, and use AI tools to trace recursive decomposition and compare multiplication counts.

MODULE III: GREEDY ALGORITHMS

Topics: Greedy algorithmic strategy applied to interval scheduling, minimum spanning trees, shortest paths, and optimal prefix coding.

AI Integration: AI tools generate scheduling scenarios and character-frequency datasets, visualize graph traversal, and compare optimal schedules, path selection, and compression ratios.

Suggested Tools: ChatGPT; VisuAlgo; Gephi; Cytoscape

Experiments:

- **Experiment-8: Activity Selection and Scheduling** Implement Activity Selection, Fractional Knapsack, and Job Scheduling, using AI tools to generate scheduling scenarios and compare optimal schedules.
- **Experiment-9: Minimum Spanning Tree and Shortest Path** Implement Prim's Algorithm, Kruskal's Algorithm, and Dijkstra's Algorithm, using Gephi and Cytoscape alongside VisuAlgo to visualize graph traversal and compare path selection.
- **Experiment-10: Huffman Encoding** Generate character frequencies, construct a Huffman Tree, and calculate the compression ratio, using AI tools to generate frequency datasets and compare compression.
-

MODULE IV: DYNAMIC PROGRAMMING

Topics: Dynamic programming formulation of classical optimization problems, including DP table construction, all-pairs shortest paths, and comparison of recursive, memoized, and tabulated implementations.

AI Integration: AI tools generate DP tables, trace solution reconstruction, visualize optimized paths, and compare route optimization along with runtime and memory trade-offs across implementation strategies.

Suggested Tools: DP Visualizer; Python Tutor; ChatGPT

Experiments:

- **Experiment-11: Dynamic Programming Table Construction** Implement Longest Common Subsequence, 0/1 Knapsack, and Edit Distance, using AI tools to generate DP tables and trace solution reconstruction.
- **Experiment-12: Shortest Path and Travelling Salesman** Implement the Floyd,Warshall algorithm and a Travelling Salesman solution, using AI tools to visualize paths and compare route optimization.
- **Experiment-13: Memorization vs Tabulation** Compare recursive, memorized, and tabulated implementations by measuring runtime and memory, using AI tools to generate execution comparisons.

MODULE V: ADVANCED TOPICS AND COMPUTATIONAL COMPLEXITY

Topics: Computationally hard problems addressed through backtracking and branch-and-bound, complexity classification (P, NP, NP-Complete, NP-Hard), and approximation algorithms.

AI Integration: AI tools visualize state-space trees, compare pruning strategies, generate reduction examples, and compare exact versus approximate solution quality.

Suggested Tools: Graphviz; Gephi; Claude; ChatGPT

Experiments:

- **Experiment-14: Backtracking and Branch-and-Bound** Implement the N Queens problem, Graph Coloring, and TSP using backtracking and branch-and-bound, using AI tools to visualize state-space trees and compare pruning.
- **Experiment-15: Complexity Classification and Approximation** Classify problems into P, NP, NP-Complete, and NP-Hard, and implement Vertex Cover Approximation and TSP Approximation, using AI tools to generate reduction examples and compare exact and approximate solutions.

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Apply asymptotic notation, amortized analysis, and recurrence solving techniques to analyse algorithm complexity.	L3 (Apply)

CO	Course Outcome Statement	Bloom's Level
CO2	Design and analyse divide-and-conquer algorithms for sorting, searching, and matrix multiplication.	L4 (Analyse)
CO3	Apply greedy algorithms to graph and optimization problems.	L3 (Apply)
CO4	Design dynamic programming solutions for classical optimization problems.	L4 (Analyse)
CO5	Apply complexity classification, backtracking, branch-and-bound, and approximation methods to computationally difficult problems.	L4 (Analyse)

CO , PO Articulation Matrix

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning
CO1	L3	3	3	2	2	3	,	,	,	1	1	,	2
CO2	L4	3	3	3	2	3	,	,	,	1	1	,	2
CO3	L3	3	2	2	2	3	,	,	,	1	1	,	2
CO4	L4	3	3	3	2	3	,	,	,	1	1	,	2
CO5	L4	3	3	2	3	3	,	,	,	1	1	,	2

Textbooks :

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*, 4th Edition, MIT Press.
2. Jon Kleinberg and Eva Tardos, *Algorithm Design*, Pearson.

Reference Books :

1. Dasgupta, S., Papadimitriou, C., and Vazirani, U., *Algorithms*.
2. Sedgewick, R., and Wayne, K., *Algorithms*.
3. Skiena, S. S., *The Algorithm Design Manual*.
4. Baase, S., and Van Gelder, A., *Computer Algorithms*.

Online Learning Resources

1. NPTEL , Design and Analysis of Algorithms , nptel.ac.in
2. MIT OpenCourseWare , Introduction to Algorithms , ocw.mit.edu
3. VisuAlgo , visualgo.net
4. Algorithm Visualizer , algorithm-visualizer.org
5. Gephi , gephi.org
6. Python Tutor , pythontutor.com

7. Wolfram Alpha , wolframalpha.com

Recommended Free & Open-Source AI Computational Tools

Visualization Tools: VisuAlgo; Algorithm Visualizer; DP Visualizer; Graphviz; Gephi; Cytoscape , used to visualize algorithm execution, recursion trees, DP tables, and graph state-space structures.
AI Assistants: ChatGPT; Claude , used to generate derivations, test cases, reduction examples, and comparative complexity analysis.
Mathematical & Tracing Tools: Wolfram Alpha; SymPy; Python Tutor , used to solve and verify recurrence relations and trace program execution step-by-step.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0510) OBJECT ORIENTED DESIGN AND PROGRAMMING LAB
(Professional Core -III Lab)
(Common to CSE & Allied Branches)

Course Objectives:

The objectives of the course is to

1. *Develop practical skills in object-oriented analysis, design, and implementation using Java programming constructs and software design principles.*
2. *Apply classes, objects, inheritance, interfaces, exception handling, collections, and multithreading to build modular and reusable applications.*
3. *Design and implement Java applications using packages, file handling, Java I/O, JDBC connectivity, and GUI development.*
4. *Utilize AI-assisted development tools for program generation, debugging, UML modelling, testing, and code optimization.*
5. *Build complete object-oriented applications following software engineering and intelligent development practices.*

List of Experiments

MODULE I: OOP FUNDAMENTALS, DATA TYPES, OPERATORS & CONTROL STRUCTURES

Topics: Primitive data types, operators, precedence, associativity, and type conversion; conditional and nested statements, switch, loops, and control-transfer statements; searching, sorting, and matrix-based applications using arrays.

AI Integration: Use ChatGPT or GitHub Copilot for program generation, test-case generation, execution tracing, and algorithm validation; use draw.io to generate flowcharts.

Suggested Tools: Java (JDK), VS Code IntelliJ IDEA, ChatGPT, GitHub Copilot, draw.io

Experiments:

1. **Experiment-1: Data Types & Operators ,** Develop Java programs to demonstrate primitive data types, operators, precedence, associativity, and type conversion.
2. **Experiment-2: Control Structures ,** Develop Java applications using conditional statements, nested conditions, switch statements, loops, and control-transfer statements.
3. **Experiment-3: Arrays, Searching & Sorting ,** Implement searching, sorting, and matrix-based applications using arrays and control structures.

MODULE II: CLASSES, OBJECTS, METHODS & COLLECTIONS

Topics: Classes, objects, constructors, constructor and method overloading, recursion; access control, nested classes, static and final members; Java Collection Framework , ArrayList, LinkedList, HashMap, HashSet, TreeMap.

AI Integration: Use StarUML and draw.io to generate UML class diagrams; use AI tools to generate class structures and recommend suitable collections.

Suggested Tools: Java (JDK), StarUML, draw.io, ChatGPT

Experiments:

1. **Experiment-4: Classes & Constructors ,** Develop Java applications using classes, objects, constructors, constructor overloading, method overloading, and recursive methods.
2. **Experiment-5: Access Control & Members ,** Design real-world object-oriented applications using access control, nested classes, static members, and final members.
3. **Experiment-6: Collection Framework ,** Implement and compare applications using the Java Collection Framework including ArrayList, LinkedList, HashMap, HashSet, and TreeMap.

MODULE III: ARRAYS, INHERITANCE & INTERFACES

Topics: One- and multi-dimensional arrays, sorting, searching, and array-manipulation techniques; inheritance , single, multilevel, hierarchical, method overriding, dynamic dispatch; interfaces, interface inheritance, abstract classes, and functional interfaces.

AI Integration: Use AI tools to generate inheritance hierarchies and compare interface-based and abstract-class-based designs.

Suggested Tools: Java (JDK), StarUML, ChatGPT, GitHub Copilot

Experiments:

1. **Experiment-7: Array Techniques ,** Develop Java programs using one-dimensional and multi-dimensional arrays, sorting, searching, and array-manipulation techniques.
2. **Experiment-8: Inheritance ,** Implement inheritance concepts including single, multilevel, and hierarchical inheritance, method overriding, and dynamic dispatch.
3. **Experiment-9: Interfaces & Abstract Classes ,** Design applications using interfaces, interface inheritance, abstract classes, and functional interfaces.

MODULE IV: PACKAGES, EXCEPTION HANDLING & JAVA I/O

Topics: User-defined packages and multi-package applications; exception handling , try-catch-finally, throw, throws, multiple catch, user-defined exceptions, try-with-resources; file handling using Scanner, byte streams, character streams, and the Java Files API.

AI Integration: Use AI tools for debugging, exception analysis, package-architecture generation, and file-operation optimization.

Suggested Tools: Java (JDK), VS Code, ChatGPT, GitHub Copilot

Experiments:

1. **Experiment-10: Packages ,** Create and implement user-defined packages and develop multi-package applications.
2. **Experiment-11: Exception Handling ,** Develop Java programs using exception-handling mechanisms including try-catch-finally, throw, throws, multiple catch blocks, user-defined exceptions, and try-with-resources.
3. **Experiment-12: File Handling ,** Implement Java file-handling applications using Scanner, byte streams, character streams, and the Java Files API.

MODULE V: STRINGS, MULTITHREADING, JDBC & JAVAFX

Topics: String, StringBuffer, StringBuilder, and string-manipulation techniques; multithreading , Thread class, Runnable interface, synchronization, producer-consumer model, deadlock simulation; JDBC connectivity with CRUD, PreparedStatement, ResultSet; JavaFX GUI design and event handling.

AI Integration: Use AI tools for multithread debugging, GUI generation, SQL generation, and intelligent code optimization.

Suggested Tools: Java (JDK), JavaFX, MySQL, ChatGPT

Experiments:

1. **Experiment-13: String Handling ,** Develop applications using String, StringBuffer, StringBuilder, and string-manipulation techniques.
2. **Experiment-14: Multithreading ,** Implement multithreaded Java applications using the Thread class, Runnable interface, synchronization, the producer-consumer model, and deadlock simulation.
3. **Experiment-15: JDBC & JavaFX Integration ,** Develop integrated Java applications using JDBC database connectivity, CRUD operations, PreparedStatement and ResultSet, and JavaFX GUI design with event handling.

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Develop Java programs using object-oriented principles, data types, operators, and control structures.	L3 (Apply)
CO2	Design and implement classes, methods, inheritance, interfaces, and collection framework-based applications.	L4 (Analyze)
CO3	Apply exception handling, packages, file operations, multithreading, and JDBC connectivity to develop robust Java applications.	L3 (Apply)
CO4	Build GUI-based applications and interactive software solutions using JavaFX and modern Java APIs.	L4 (Analyze)
CO5	Utilize AI-assisted software development tools for UML modelling, code generation, testing, debugging, and application enhancement.	L4 (Analyze)

CO , PO Articulation Matrix

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning	PSO1 Computing Core	PSO2 Profnl Dev.
CO1	L3	3	2	2	1	3	,	,	,	1	1	,	2	3	2
CO2	L4	3	3	3	2	3	,	,	,	1	1	1	2	3	2
CO3	L3	3	3	2	2	3	,	,	,	1	1	1	2	3	2
CO4	L4	3	2	3	2	3	,	,	,	1	2	1	2	3	2
CO5	L4	3	2	3	2	3	,	,	1	2	2	2	3	3	3

Textbooks :

1. Seth, A., and Juneja, B. L., *JAVA One Step Ahead*, Oxford University Press.
2. Samanta, D., and Sarma, M., *Joy with JAVA: Fundamentals of Object-Oriented Programming*, Cambridge University Press, 2023.
3. Deitel, P., and Deitel, H., *Java 9 for Programmers*, 4th Edition, Pearson Education.

Reference Books :

1. Schildt, H., *The Complete Reference Java*, 11th Edition, McGraw-Hill.
2. Liang, Y. D., *Introduction to Java Programming*, Pearson Education..

Online Learning Resources

1. NPTEL , Programming in Java , nptel.ac.in
2. Oracle Java Documentation , docs.oracle.com/en/java
3. StarUML Documentation , staruml.io
4. draw.io Documentation , draw.io
5. GitHub Copilot Documentation , github.com/features/co-pilot
6. https://www.youtube.com/@JoyofLearningCSE/featured

Recommended Free & Open-Source Java AI Tools

Language & IDE: Java (JDK), VS Code, IntelliJ IDEA , development and debugging.
UML & Diagrams: StarUML, draw.io , class, inheritance and flow diagrams.
Database & GUI: MySQL, JavaFX (Scene Builder) , JDBC connectivity and interfaces.
AI Assistants: ChatGPT, GitHub Copilot , code generation, debugging, test generation and optimization.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY:: PUTTUR
(AUTONOMOUS)

II B.Tech – I Sem.

L	T	Pr	P	C
2	0	0	0	2

(26CE0107) SUSTAINABILITY AND GREEN ENGINEERING
(Common to all branches)

Prerequisite: None

Course Objectives:

The objectives of this course is

1. *Understand the concepts of sustainability, carbon cycle, and the environmental impact of engineering activities.*
2. *Evaluate sustainable materials, their life cycle, and environmental performance.*
3. *Apply energy calculations including embodied energy, operational energy, and life cycle energy.*
4. *Assess green standards, energy codes, and performance rating systems (LEED, GRIHA).*
5. *Integrate AI/ML tools for carbon footprint analysis, energy optimization, and sustainable design decision-making.*

UNIT-I :	**Fundamentals of Sustainability and Carbon Cycle**
Content	Definition of sustainability: three pillars (economic, social, environmental); Brundtland definition; Sustainable Development Goals (SDGs): SDGs 6, 7, 9, 11, 12, 13. Carbon cycle: biotic and abiotic components; Greenhouse gases: CO2, CH4, N2O, HFCs; Carbon dioxide equivalent (CO2e). Carbon footprint: definition, calculation methodology, carbon accounting; Climate change: global warming, sea level rise, extreme events.
AI Integration	AI for carbon footprint tracking: overview of ML-based life cycle assessment tools. Python-based carbon footprint calculator computing embodied carbon from material/process quantities. Free/Open-Source AI Tools: openLCA; Python (Pandas, Matplotlib) on Google Colab; Global Carbon Atlas.

UNIT-II :	**Sustainable Materials and Resource Selection**
Content	Sustainable materials and resource efficiency; Indoor air quality considerations; Recycled and alternative material sources; Durability and service life assessment. Life cycle of engineering materials; Multi-criteria decision analysis for material selection.
AI Integration	ML for sustainable material selection using multi-criteria decision analysis in Python. AI-based prediction of material performance using

	regression models (scikit-learn). Comparative LCA using openLCA. Free/Open-Source AI Tools: scikit-learn; openLCA; Python (NumPy, scikit-learn) on Google Colab.

UNIT-III :	**Embodied Energy and Operational Energy**
Content	Definition of embodied energy: extraction, processing, manufacturing, transport, construction, demolition; Operational energy: heating, cooling, lighting, equipment use. Life cycle energy: sum of embodied and operational energy; Energy concept: primary energy, renewable vs. non-renewable energy. Energy balance: reducing operational energy vs. increasing efficiency investment (embodied energy trade-off).
AI Integration	Python-based embodied energy calculator. AI for energy optimization: overview of EnergyPlus and OpenStudio simulations. ML for operational energy prediction (Linear Regression, Random Forest using scikit-learn). Free/Open-Source AI Tools: EnergyPlus; OpenStudio; Python (scikit-learn, NumPy) on Google Colab.

UNIT-IV :	**Green Rating Systems and Energy Codes**
Content	Energy Conservation and Sustainability Building Code (ECSBC-2023): scope, mandatory and prescriptive requirements; Green building rating systems: LEED, GRIHA, IGBC, BEE star rating. Passive design strategies: orientation, natural ventilation, daylighting; Net Zero Energy concepts: definition, pathways.
AI Integration	AI for automated LEED/GRIHA compliance checking. ML for performance prediction from design parameters. IoT sensors for real-time energy monitoring; AI-driven optimization case studies. Free/Open-Source AI Tools: EnergyPlus; OpenStudio; Python for data analysis on Google Colab.

UNIT-V :	**Circular Economy and Environmental Sustainability**
Content	Circular economy principles: reduce, reuse, recycle, recover, redesign; Waste hierarchy; Industrial waste utilization. Water recycling; Green procurement policies; Carbon sequestration potential of green infrastructure; Acid rain: causes, effects, prevention.
AI Integration	AI for waste prediction using ML models. Computer vision for automated waste sorting (overview). Python for waste audit data analysis. Sustainable infrastructure planning using AI and GIS. Free/Open-Source AI Tools: Python (scikit-learn, Pandas) on Google Colab; QGIS; openLCA; i-Tree.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Remember and explain the principles of sustainability, SDGs, the carbon cycle, and greenhouse gas accounting.	**L1–L2**
CO2	Apply knowledge of sustainable materials to evaluate their environmental performance using life cycle assessment methods.	**L3 (Apply)**
CO3	Analyze the embodied energy and operational energy of engineering systems using computational tools.	**L4 (Analyze)**
CO4	Evaluate green rating systems and assess compliance of design with energy codes using simulation and AI tools.	**L4–L5**
CO5	Design sustainable engineering solutions integrating circular economy principles and AI-driven optimization tools.	**L5–L6**

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1/L2	Remember/Understand	**CO1**	SDG identification; carbon cycle and greenhouse gas concept recall.
L3	Apply	**CO2**	Material LCA evaluation; sustainable material selection exercises.
L4	Analyze	**CO3, CO4**	Embodied/operational energy calculation; green rating compliance analysis.
L5/L6	Evaluate/Create	**CO5**	Sustainable engineering solution design with AI-driven optimization.

CO – PO / PSO Articulation Matrix :

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	**2**	1	–	–	–	**3**	**3**	**2**	–	**2**	–	**2**	**2**	**2**
CO2	**2**	**2**	**2**	**2**	**2**	**2**	**3**	**2**	1	**2**	–	**2**	**2**	**2**
CO3	**2**	**2**	**2**	**2**	**3**	**2**	**3**	1	1	**2**	1	**2**	**2**	**3**
CO4	**2**	**2**	**3**	**2**	**3**	**3**	**3**	**2**	1	**2**	1	**3**	**2**	**3**
CO5	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**3**	**2**	**3**

CO-PO Correlation	CO4 and CO5 show strong correlation with PO3 (Design) and PO5 (Modern Tools) through green rating compliance and AI-driven

Highlights	sustainable design. All COs map strongly to PO7 (Environment & Sustainability), reflecting the course's core focus. CO5 additionally connects to PO10 and PO12 through integrative project-based sustainable solution design.

Textbooks :

1. Kibert, C. J., *Sustainable Construction: Green Building Design & Delivery*, 4th Ed., Wiley, 2016.
2. Goodhew, S., *Sustainable Construction Processes*, Wiley-Blackwell, 2016.

Reference Books :

1. Langston, C. A., & Ding, G. K. C., *Sustainable Practices in the Built Environment*, Butterworth-Heinemann, 2011.
2. Bureau of Energy Efficiency, *Energy Conservation and Sustainability Building Code (ECSBC 2024)*, Government of India, 2024.
3. Goswami, D., Kreith, F., & Kreider, J., *Principles of Solar Engineering*, Taylor & Francis, 2000

Online Resources :

1. NPTEL: Sustainable Development – https://nptel.ac.in/courses/105105157
2. Bureau of Energy Efficiency (BEE), India: https://beeindia.gov.in
3. IGBC (Indian Green Building Council): https://igbc.in
4. openLCA (free LCA software): https://www.openlca.org
5. EnergyPlus (free building energy simulation): https://energyplus.net

Web Resources & Nptel Links :

1. NPTEL: Sustainable Development – https://nptel.ac.in/courses/105105157
2. NPTEL: Green Buildings https://archive.nptel.ac.in/courses/105/102/105102195
3. Bureau of Energy Efficiency (BEE), India: https://beeindia.gov.in
4. IGBC (Indian Green Building Council): https://igbc.in
5. openLCA (free LCA software): https://www.openlca.org
6. EnergyPlus (free building energy simulation): https://energyplus.net
7. OpenStudio (free building energy modelling): https://openstudio.net
8. Global Carbon Project: https://www.globalcarbonproject.org

Recommended Free & Open-Source AI/Computational Tools

1. **openLCA (free, open-source)** – Life Cycle Assessment software for embodied carbon calculation
2. EnergyPlus (free, US DOE) – Building energy simulation for operational energy analysis
3. OpenStudio (free, NREL) – Building energy modeling interface for EnergyPlus
4. Python (scikit-learn, Pandas, NumPy) on Google Colab – ML for energy prediction and material selection
5. QGIS (free) – Green infrastructure planning and spatial sustainability analysis

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , I Sem.

L	T	Pr	P	C
1	0	0	2	2

(26CS4770) AI ASSISTED APPLICATION DEVELOPMENT
(Skill Enhancement Course- I)
(Common to CSE & Allied branches)

Course Objectives:

The objectives of the course is to

1. *Introduce students to modern web application development using frontend, backend, and database technologies.*
2. *Enable students to utilize AI-assisted development tools for designing, developing, testing, debugging, and improving applications.*
3. *Develop skills to create responsive and interactive web interfaces using HTML, CSS, and JavaScript.*
4. *Build competency in developing full-stack web applications through integration of client-side, server-side, and database components.*
5. *Promote efficient, collaborative, and responsible software development practices using AI-enabled workflows.*

List of Experiments

Module I – HTML and CSS: Web Interface Design (Weeks 1–3)

1. AI-Assisted Web Page Design using HTML | HTML5, VS Code

Develop a personal portfolio website using HTML5 semantic elements — header, nav, section, article, footer — incorporating text, images, hyperlinks, ordered/unordered lists, tables, and embedded multimedia (video/audio). Use GitHub Copilot or ChatGPT to generate the initial page structure; critically review and refine the AI output before finalising. Validate the HTML using the W3C Validator; document all changes made to the AI-generated code and explain each modification.

2. Responsive Web Forms using HTML and CSS | HTML5, CSS3

Design a responsive registration and login form with appropriate input controls (text, email, password, radio, checkbox, select, file, submit) and a styled validation interface. Use an AI tool (GitHub Copilot or ChatGPT) to generate form layout and CSS; compare the AI output with a manually designed version and document the differences. Apply media queries to make the form responsive on desktop and mobile viewports (test at 320px, 768px, 1200px widths).

3. CSS Styling and Responsive Layout Development | CSS3

Develop responsive webpages using CSS selectors (class, ID, attribute, pseudo-class, pseudo-element), typography, colour schemes, the Box Model (margin, border, padding, content), Flexbox (flex-direction, justify-content, align-items, flex-wrap), and CSS Grid

(grid-template-columns, grid-template-rows, grid-area). Generate an initial layout using AI assistance; document the properties that were accepted, modified, or rejected. Compare the AI-generated layout with a manually coded version and report differences in code quality and responsiveness.

Module II – CSS Components and JavaScript Fundamentals (Weeks 4–6)

4. User Interface Component Development | HTML5, CSS3

Create reusable UI components: a responsive navigation bar (hamburger menu for mobile, horizontal links for desktop), a card grid (image, title, description, button), a responsive dashboard (stats panel, sidebar, content area), and a dropdown menu. Use AI tools to generate component CSS and HTML; evaluate each component for UI consistency, accessibility (alt text, ARIA labels, colour contrast), and responsiveness. Document the AI-generated code, the improvements made, and the rationale for each change.

5. JavaScript Programming Fundamentals | JavaScript ES6

Develop interactive web pages demonstrating: variables (let, const, var — scope differences), operators, control flow (if-else, switch, ternary), loops (for, while, for-of, for-in), functions (declaration, expression, arrow function), arrays (map, filter, reduce, find, forEach), and objects (properties, methods, destructuring, spread operator). Use GitHub Copilot to generate starter code for each concept; test each snippet in the browser console and document cases where the AI-generated code contained errors or inefficiencies.

6. DOM Manipulation and Event Handling | JavaScript, Browser Console

Develop an event-driven to-do list application using DOM operations: select elements (getElementById, querySelector, querySelectorAll), create and append elements (createElement, appendChild, insertBefore), modify content (innerHTML, textContent, setAttribute), and handle events (click, keypress, change, submit, mouseover). Implement dynamic content updates — add task, mark complete, delete task, filter by status. Use an AI tool to generate the event listener logic; identify and fix at least 2 bugs in the AI-generated code before submission.

Module III – Client-Side Advanced and Backend Setup (Weeks 7–9)

7. Client-Side Form Validation and Browser Storage | JavaScript

Implement client-side form validation for a registration form: validate name (non-empty, min 3 chars), email (regex pattern), password (min 8 chars, at least one number and special character), confirm password (match check), and phone (10 digits). Display real-time error messages below each field without page reload. Store validated form data in LocalStorage (persistent) and SessionStorage (session-only); retrieve and display stored data on page load. Use AI to generate the validation regex patterns; test each against 5 valid and 5 invalid inputs and document the results.

8. API Integration and Dynamic Content Generation | JavaScript, REST APIs

Consume two external public APIs (OpenWeatherMap — current weather by city; REST Countries — country information by name) using the Fetch API and async/await. Parse and display JSON responses dynamically on the webpage — weather card (temperature, humidity, icon, description) and country card (flag, capital, population, languages).

Handle API errors (network error, invalid city, 404 response) with user-friendly messages. Use ChatGPT to generate the Fetch API code skeleton; test with 5 valid and 3 invalid inputs and verify error handling.

9. Backend Application Setup using Node.js and Express.js | Node.js, Express.js

Set up a Node.js and Express.js backend server: initialise a project (npm init), install Express (npm install express), create a server (app.js) with app.listen() on port 3000, define 5 routes (GET /, GET /about, GET /students, POST /students, DELETE /students/:id) with appropriate HTTP methods and response JSON. Implement middleware for request logging (log method, URL, timestamp for every request) and CORS. Use GitHub Copilot to generate the Express boilerplate; test all routes using Postman; document the AI-generated code and all manual additions.

Module IV – REST API, Database, and Full-Stack Integration (Weeks 10–12)

10. REST API Development and CRUD Operations | Express.js, Postman

Develop a fully functional RESTful API for a Student resource with all four CRUD operations: POST /students (create — validate required fields), GET /students (read all — support query param filtering by department), GET /students/:id (read one — return 404 if not found), PUT /students/:id (update — partial update supported), DELETE /students/:id (delete — return 204 on success). Use proper HTTP status codes for all responses (200, 201, 204, 400, 404, 500). Use GitHub Copilot to generate route handlers; test all endpoints in Postman with 5 test cases per route; export and submit the Postman collection.

11. Database Integration and Authentication | MongoDB, Node.js, Express.js

Connect the Express.js backend to MongoDB Atlas using Mongoose: define a Student schema (name, roll_no, email, department, marks — with type, required, and validation rules); implement the CRUD API from Week 10 using Mongoose model methods (save, find, findById, findByIdAndUpdate, findByIdAndDelete). Implement user authentication: register endpoint (hash password using bcrypt, store in MongoDB), login endpoint (verify password, generate JWT token using jsonwebtoken), and a protected /profile route (verify JWT using middleware). Use AI to generate the Mongoose schema and bcrypt/JWT code; test with 5 user registrations and login scenarios.

12. Frontend–Backend Integration | React.js, REST APIs

Create a React.js frontend (using Create React App or Vite) with 4 components: StudentList (fetch and display all students from GET /students), StudentForm (add a new student via POST /students with controlled form inputs), StudentCard (display individual student data with edit and delete buttons), and a Navbar. Connect to the Express.js + MongoDB backend from Week 11 using Axios (axios.get, axios.post, axios.put, axios.delete). Handle loading states and error states in each component. Use GitHub Copilot to scaffold components; identify and fix at least 2 integration issues encountered during testing.

Module V – Testing, Deployment, and Mini Project (Weeks 13–15)

13. Full-Stack Application Testing and Optimization | React.js, Node.js, Jest, Lighthouse

Write unit tests for 3 backend API routes using Jest and Supertest (test POST /students with valid data, invalid data, and duplicate entry). Write 2 React component tests using

React Testing Library (render StudentList and verify it displays fetched students; render StudentForm and simulate a form submission). Run Google Lighthouse on the frontend (target: Performance > 80, Accessibility > 90, Best Practices > 90); document issues found and apply 3 optimisations (image compression, lazy loading, unused CSS removal). Use ChatGPT to generate test case descriptions; write the actual test code yourself.

14. Deployment and Application Monitoring | Render / Vercel / Railway, GitHub

Deploy the full-stack application: push the codebase to a GitHub repository with a clear README (project description, setup instructions, API documentation, screenshots). Deploy the React frontend to Vercel (connect GitHub repo, set build command and output directory). Deploy the Node.js + Express.js backend to Render or Railway (set environment variables for MongoDB Atlas URI and JWT secret). Test the deployed application end-to-end: register a user, login, add 3 students, update 1, delete 1. Use AI tools to generate the README and troubleshoot any deployment errors; document each error encountered and how it was resolved.

15. AI-Assisted Web Application Mini Project | MERN Stack (MongoDB, Express.js, React.js, Node.js)

Design and develop a complete web application (choose from: Student Management System, Smart Attendance System, Event Registration Portal, Expense Tracker, or Task Management Dashboard) integrating all skills from Weeks 1–14. The project must include: (a) a React.js frontend with minimum 5 components and responsive CSS; (b) a Node.js + Express.js REST API with minimum 6 routes covering full CRUD; (c) MongoDB with Mongoose for data persistence and at least 2 schemas; (d) JWT-based authentication with protected routes; (e) deployment on a public URL with GitHub repository; (f) AI usage documentation — a written report listing every AI tool used, the prompts given, the output received, what was accepted, modified, and rejected. Present the working application in a 10-minute demo.

AI Tools Integration (Lab)	Primary coding tools: GitHub Copilot (VS Code extension — autocomplete and inline suggestions), ChatGPT / Claude (code generation, debugging, explanation), Microsoft Copilot (integrated in Edge browser and Office). For UI design: Figma AI (layout suggestions), v0 by Vercel (React component generation from descriptions). For testing: ChatGPT (generating test case scenarios). For deployment: GitHub Copilot CLI (command-line assistance). Responsible AI use policy: all AI-generated code must be reviewed, tested, and understood before submission; a written AI usage log (tool used, prompt, output, what was changed) must accompany every session submission. Copying AI output without testing or understanding is not acceptable.

Course Outcomes (COs

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Design and develop responsive web interfaces using HTML and CSS with AI-assisted development tools. (L6)	L6 (Create)

CO2	Develop interactive client-side applications using JavaScript and AI-assisted coding and debugging techniques. (L6)	L6 (Create)
CO3	Develop server-side applications and integrate databases for web application development. (L6)	L6 (Create)
CO4	Build, test, and integrate full-stack web applications using AI-assisted development workflows. (L6)	L6 (Create)
CO5	Deploy and maintain web applications while applying responsible and effective use of AI tools in software development. (L5)	L5 (Evaluate)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Describe HTML5 elements; explain CSS Box Model; define REST API methods.
L3	Apply	**CO1–CO3**	Build responsive forms; implement DOM event handlers; set up Express.js server.
L4	Analyse	**CO3, CO4**	Analyse REST API error categories; compare Mongoose schema validation strategies.
L5	Evaluate	**CO5**	Evaluate Lighthouse audit results; assess AI tool usage responsibility.
L6	Create	**CO1–CO4**	Design and implement a full MERN stack web application with JWT authentication.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	P M
CO1	L6	**3**	**2**	**3**	1	**3**	**2**	–	1	1	**2**	
CO2	L6	**3**	**3**	**3**	**2**	**3**	1	–	1	1	**2**	
CO3	L6	**3**	**3**	**3**	**2**	**3**	1	–	1	1	**2**	
CO4	L6	**3**	**3**	**3**	**3**	**3**	**2**	–	1	**2**	**2**	
CO5	L5	**3**	**2**	**2**	**2**	**3**	**2**	1	**2**	**2**	**2**	

CO-PO Correlation Highlights	Strong PO3 (Design/Development) through full-stack MERN application design across all sessions. PO5 through GitHub Copilot, ChatGPT, v0 by Vercel, Jest, Lighthouse, Postman, and Render/Vercel deployment. PO6 and PO8 through responsible AI use policy, AI usage log requirement, and web accessibility standards.

LEARNING RESOURCES

Textbooks

1. TB1: Jon Duckett, HTML and CSS: Design and Build Websites, Wiley Publications.
2. TB2: David Flanagan, JavaScript: The Definitive Guide, 7th Edition, O'Reilly Media.
3. TB3: Vasan Subramanian, Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React and Node, Apress.

Reference Books

1. Ethan Brown, Web Development with Node and Express, 2nd Edition, O'Reilly Media.
2. Robert W. Sebesta, Programming the World Wide Web, 8th Edition, Pearson Education.
3. Andrew Ng, Generative AI for Everyone, DeepLearning.AI.

Online Resources

1. MDN Web Docs (HTML, CSS, JS) – developer.mozilla.org | W3Schools – w3schools.com
2. React Documentation – react.dev | Node.js Docs – nodejs.org/docs | Express.js – expressjs.com
3. MongoDB Docs – mongodb.com/docs | Mongoose – mongoosejs.com
4. GitHub Skills – skills.github.com | VS Code Docs – code.visualstudio.com/docs
5. FreeCodeCamp – freecodecamp.org | OpenAI Developer Resources – platform.openai.com/docs

Suggested Mini Projects:

1. Student Management System
2. Smart Attendance System
3. Event Registration Portal
4. Expense Tracker
5. Inventory Management System
6. Task Management Dashboard
7. Online Quiz Application
8. Campus Information Portal

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech – I Sem.

L	T	Pr	P	C
2	0	0	0	0

(26HS0826) TECHNICAL PAPER WRITING & IPR
(Common to all branches)

Prerequisites: Communicative English

Course Objectives :

The objectives of this course is to

1. *Distinguish technical communication from general communication and identify the types of technical documents used in engineering.*
2. *Develop competency in structuring formal technical reports including abstracts, literature reviews, and data presentation.*
3. *Apply scientific writing conventions for research proposals and IMRAD-structured papers.*
4. *Understand the fundamentals of Intellectual Property Rights including patents, trademarks, and copyrights.*
5. *Draft patent applications and develop IP strategy awareness using AI-assisted writing and patent search tools.*

UNIT-I :	**Foundations of Technical Communication**
Content	Technical Communication vs. General Communication: Precision, objectivity, and audience-awareness; Types of Technical Documents: Reports, Proposals, Manuals, Specifications, Abstracts.Writing Process for Technical Documents: Planning, Drafting, Revising, Proofreading; Audience Analysis: Primary and secondary audiences.Language in Technical Writing: Active vs. Passive voice, Nominalisations, Clarity and conciseness.
AI Integration	AI Tool: Grammarly (free) to analyse a technical paragraph for passive voice overuse. AI Tool: Hemingway Editor to improve readability to Grade 10 or below. Activity: compare before/after AI-edited document versions.

UNIT-II :	**Technical Report Writing**
Content	Structure of a Formal Report: Title page, Abstract, Table of Contents, Introduction, Methodology, Results, Discussion, Conclusion, References, Appendices. Abstract Writing: Informative vs. Descriptive abstracts; Literature Review: Searching, evaluating, synthesising sources. Data Presentation: Tables, Figures, Charts; Citation and Referencing: IEEE, APA, and MLA styles; avoiding plagiarism.

AI Integration	AI Tool: Elicit.org for AI-powered literature search. AI Tool: Zotero for citation management and bibliography generation. AI Tool: ChatGPT for first-draft abstract generation. Plagiarism check using Turnitin or Copyleaks.

UNIT-III :	**Research Proposals and Scientific Writing**
Content	Research Proposal Structure: Background, Problem Statement, Objectives, Methodology, Timeline, Budget; Scientific Paper Format: IMRAD Structure. Writing Results vs. Discussion: Factual reporting vs. Interpretation; Peer Review Process: blind review, revision, resubmission; Conference Papers vs. Journal Articles.
AI Integration	AI Tool: Semantic Scholar for open-access paper search and IMRAD structure analysis. AI Tool: Writefull for AI feedback on scientific language. Activity: write a research proposal with AI structural review.

UNIT-IV :	**Intellectual Property Rights — Fundamentals**
Content	Introduction to IPR: Definition, importance, and types Patents, Trademarks, Copyrights, Trade Secrets, Geographical Indications. Patent System in India: Indian Patent Act 1970, Patent Office, application process; Patentability Criteria: Novelty, Inventive Step, Industrial Applicability. Patent Search: Prior art search using databases; Copyright in Engineering: Software copyright, Open-source licenses (GPL, MIT, Apache).
AI Integration	AI Tool: Google Patents for prior art search. AI Tool: Lens.org for patent document search and analysis. Activity: identify and analyse patent claims in a chosen engineering domain.

UNIT-V :	**Patent Drafting and IP Strategy**
Content	Structure of a Patent Application: Title, Abstract, Background, Summary, Detailed Description, Claims, Drawings; Types of Claims: Independent and Dependent claims. Filing a Provisional Patent Application (PPA) in India; IP Strategy for Startups and Engineers: trade secrets vs. patents, licensing, technology transfer. Case Studies: Famous patent disputes (Apple vs. Samsung, Monsanto seed patents, pharmaceutical patents).

AI Integration	AI Tool: ChatGPT to draft patent claims for a mechanical invention and critique for clarity. AI Tool: IP India Online Portal navigation. Capstone: write a complete patent disclosure document.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and understand the principles, types, and structure of technical reports, and explain the fundamentals of Intellectual Property Rights.	**L1–L2**
CO2	Apply the principles of technical writing to produce well-structured laboratory reports, project reports, and technical proposals.	**L3 (Apply)**
CO3	Analyse technical documents for clarity, coherence, accuracy, and compliance with formatting standards using AI-powered review tools.	**L4 (Analyse)**
CO4	Evaluate patent documents, prior art, and IP claims to assess novelty and inventiveness of an engineering innovation.	**L5 (Evaluate)**
CO5	Create a complete technical report and draft a patent disclosure for an engineering project, integrating AI tools for writing assistance.	**L6 (Create)**

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1/L2	Remember Understand	**CO1**	IPR type identification; technical document structure recall.
L3	Apply	**CO2**	Report drafting exercises using AI writing tools; proposal structuring.
L4	Analyse	**CO3**	AI-assisted document clarity and compliance review.
L5/L6	Evaluate/Create	**CO4, CO5**	Patent claim evaluation; patent disclosure document creation.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO2
CO1	1	1	1	1	1	1	1	**2**	1	**3**	**2**	**2**	**2**	**2**
CO2	**2**	**2**	**2**	**2**	**2**	1	1	**2**	**2**	**3**	**2**	**3**	**2**	**2**
CO3	**2**	**3**	**2**	**2**	**3**	1	1	**2**	**2**	**3**	**2**	**3**	**2**	**2**
CO4	**2**	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**	**2**	**2**	**3**	**3**	**3**

CO5	**2**	**2**	**3**	**2**	**3**	1	1	**2**	**3**	**3**	**3**	**3**	**3**	**3**
CO-PO Correlation Highlights	CO3 and CO4 strongly align with PO2 (Problem Analysis) and PO8 (Ethics) through document analysis and patent evaluation. CO5 shows the broadest correlation, mapping to PO3, PO9, PO10, PO11, and PO12 through the integrative report-and-patent creation capstone. PO10 (Communication) is consistently addressed across all COs as the course's core competency.													

Textbooks:

1. Markel, M., & Selber, S. (2022). Technical Communication (13th ed.). Bedford/St. Martin's.
2. Ahuja, B. N., & Ahuja, S. S. (2021). Intellectual Property Rights: Patents, Trade Marks, Copyrights and Related Topics (4th ed.). Khanna Publishers.

Reference Books :

1. Lannon, J., & Gurak, L. (2020). Technical Communication (15th ed.). Pearson Education.
2. Anderson, P. V. (2019). Technical Communication: A Reader-Centred Approach (9th ed.). Cengage.
3. Meganathan, R. (2019). Intellectual Property Rights for Engineers. Universities Press (India).
4. Cornish, W. R., & Llewelyn, D. (2019). Intellectual Property: Patents, Copyright, Trademarks and

Online Resources

- IP India – Patent Office: www.ipindia.gov.in
- Google Patents: patents.google.com
- Lens.org – Free Patent and Scholar Search: www.lens.org
- WIPO – Patent Resources: www.wipo.int
- Purdue OWL – Technical Writing: owl.purdue.edu

YouTube / Video Resources

- YouTube: NPTEL – Technical English for Engineers – www.youtube.com/c/nptel
- YouTube: IP India – Patent Filing Guide – search on YouTube
- YouTube: WIPO – Intellectual Property for Engineers – www.youtube.com/@WIPOChannel
- YouTube: How to Write a Patent Application – PatentFile – search on YouTube
- YouTube: Scholarly Writing Tips – Academic English – search on YouTube

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , II Sem.

L	T	Pr	P	C
2	0	0	0	2

(26ME0305) MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS
(Common to all branches)

PREREQUISITES: Basic Economics, Basic Mathematics, Statistics

COURSE OBJECTIVES

The objectives of this course is to

1. Understand the fundamental concepts of managerial economics, demand analysis, and elasticity to make data-driven business decisions.
2. Apply production and cost analysis techniques, including break-even analysis, to optimize organizational resource allocation.
3. Analyse various forms of business organizations, market structures, and pricing strategies for competitive business environments.
4. Evaluate capital budgeting proposals using NPV, IRR, ARR, and Payback methods for strategic investment decision-making.
5. Prepare and interpret financial statements, compute key financial ratios, and employ AI/ML techniques for automated financial analysis.

UNIT – I	**UNIT I: INTRODUCTION TO MANAGERIAL ECONOMICS**
Core Content	Managerial Economics – Meaning, Nature, Scope and significance for Engineers – Role of Managerial Economics in Business Decision-Making; Relationship with Economics, Management, Mathematics, Statistics, and Accounting; Economic Analysis - Micro vs. Macroeconomics; positive vs. normative economics and the Decision-Making Process. - Demand Analysis – Concept, Determinants, Demand Function, Law of Demand and its exceptions. - Elasticity of Demand – Types, Measurement, Business Applications; Demand Forecasting – Meaning, Need, Methods**.**
AI Integration	• AI Tool: ChatGPT – generate demand scenarios for an engineering product (e.g., solar panels) and interpret price elasticity • AI Tool: Google Sheets + FORECAST function – implement Moving Average and Exponential Smoothing for a sample sales dataset • Activity: Use publicly available CMIE or RBI data; apply AI-assisted regression in Google Colab to forecast demand

UNIT – II	**PRODUCTION AND COST ANALYSIS**
Core Content	Production Concepts and Modern Manufacturing Systems - Production Function – Isoquants and Isocosts; Least-Cost Combination of Inputs, MRTS ; Cost Concepts – Fixed, Variable, Total, Average, Marginal Costs; Break-

	Even Analysis – Concept, Assumptions, Applications; Determination of Break-Even Point; Margin of Safety and Profit Planning(Simple problems).
AI Integration	• AI Tool: Google Sheets / LibreOfficeCalc (free) – build a BEA spreadsheet with dynamic charts • AI Tool: ChatGPT – generate a cost function for a manufacturing scenario and identify optimal production level • Activity: Collect cost data of a local small enterprise; perform AI-assisted CVP analysis and present findings • Data Tool: Plotly (open-source Python) or Google Data Studio – visualise cost curves..

UNIT– III	**BUSINESS ORGANISATIONS AND MARKET STRUCTURES**

Core Content	Forms of Business Organisations – Sole Proprietary, Partnership, Joint Stock Companies, Startups and Digital Enterprises; Public vs Private Sector in Digital Economy; Market Structures – Classification of Markets with real-world examples - Pricing Methods and Strategies.
AI Integration	• AI Tool: ChatGPT – simulate a duopoly pricing game and find Nash Equilibrium with AI assistance • Activity: Analyse pricing strategies of three competing tech products using online data and AI classification • Discussion: How do AI algorithms determine prices? Dynamic pricing in e-commerce (Amazon, Uber Surge Pricing)

UNIT– IV	CAPITAL BUDGETING

Core Content	Capital -Introduction, Meaning, Nature and Significance - Capital Budgeting – Concept, Nature, Significance in Modern Enterprises, Role in Strategic Decision-Making; Methods – Payback Method, Accounting Rate of Return (ARR), Net Present Value (NPV), Internal Rate of Return (IRR)- Profitability Index (PI).
AI Integration	• AI Tool: Google Sheets + NPV/IRR functions – evaluate 3 capital investment alternatives • Monte Carlo simulation for capital budgeting risk analysis • AI-based working capital optimization using predictive models • Decision tree analysis for investment appraisal using scikit-learn.

UNIT– V	FINANCIAL ACCOUNTING AND ANALYSIS

Core Content	Accounting Principles and Digital Accounting Systems; Journal, Ledger, Trial Balance; Final Accounts – Trading Account, Profit and Loss Account, Balance Sheet with Simple Adjustments; Financial Analysis – Liquidity Ratios, Activity Ratios, Capital Structure Ratios, Profitability Ratios..
AI Integration	• Automated financial statement preparation using Python (pandas) • AI Tool: ChatGPT – interpret a company's financial ratios from a pasted balance sheet and income statement • Machine learning for financial ratio prediction and company health

	assessment • Project: Download annual report of an Indian engineering company (BHEL, L&T, Infosys) and perform ratio analysis

AI TOOLS USED IN THIS COURSE

S.No.	AI Tool	Application in Course
1	ChatGPT / Claude AI	Concept clarification, case study generation, problem solving assistance
2	Microsoft Copilot	Financial analysis, spread sheet automation, data interpretation
3	Google Gemini	Research support, market structure comparisons, summarisation
4	Tally AI / Zoho Books AI	Digital accounting simulations, ledger preparation, journal entries
5	Statista / Tableau AI	Data visualisation for demand trends and financial ratios
6	Wolfram Alpha	Quantitative problems – NPV, IRR, Break-Even, elasticity calculations
7	Grammarly AI	Report writing, case study documentation, assignments

COURSE OUTCOMES (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and understand fundamental concepts of economics including demand, supply, market structures, and the time value of money.	L1 – Remember / L2 – Understand
CO2	Apply economic theories of demand forecasting, cost analysis, and production functions to engineering project decision-making scenarios.	L3 – Apply
CO3	Analyse financial statements, cost-volume-profit relationships, and capital budgeting decisions using traditional and AI-powered financial tools.	L4 – Analyse
CO4	Evaluate investment alternatives using NPV, IRR, and Payback Period methods and assess business viability under uncertainty using scenario analysis.	L5 – Evaluate
CO5	Create a comprehensive mini-project business plan incorporating market analysis, cost estimation, financial projections, and AI-assisted forecasting.	L6 – Create

CO-PO MAPPING MATRIX

Scale: 3 = High, 2 = Medium, 1 = Low, '-' = No correlation

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	1	1	1	2	1	1	1	2	2	2	1	1
CO2	2	3	2	2	2	2	1	1	1	2	3	2	2	2
CO3	2	3	2	3	3	2	1	1	2	2	3	3	2	2
CO4	2	3	2	3	3	2	1	1	2	2	3	3	2	2
CO5	2	3	3	3	3	2	2	1	3	3	3	3	3	3

LEARNING RESOURCES

TEXT BOOKS

1. Mithani D.M, *Managerial Economics: Theory and Applications*, HimalayaPublishing House, 10th Edition, 2025.
2. Pandey, I.M, *Financial Management*, Vikas Publishing House, New Delhi, 12th Edition, 2022.
3. Chandramouli Subramanian &Asha A George, *Artificial Intelligence: Principles and Applications*, Universities Press Private Limited, HYD, First Edition, 2026.

REFERENCES

1. Aryasri A.R, *Managerial Economics and Financial Analysis*, McGraw-Hill Education, Revised Edition, 2020.
2. Prasanna Chandra, *Financial Management: Theory and Practice*, Tata McGraw-Hill, New Delhi, 10th Edition, 2019
3. Kumar U.D, *Business Analytics: The Science of Data-Driven Decision Making*, Wiley 2nd Edition, 2021.
4. Siddiqui, S. A., *Engineering Economics and Financial Accounting*, S. Chand Publishing, New Delhi, 3rd Edition, 2019.
5. Gupta, G. S., *Economics for Engineers*, Tata McGraw-Hill Education, 2nd Edition, 2017.

ONLINE RESOURCES:

- NPTEL (IIT), Managerial Economics**,** https://nptel.ac.in/courses/110105063
- NPTEL (IIT), Financial Accounting for Engineers, https://nptel.ac.in/courses/110107114
- Coursera**,** Economics for Non-Economists (Univ. of California), https://www.coursera.org/learn/economics
- edX**,** Financial Accounting (MIT OpenCourseWare), https://www.edx.org/course/financial-accounting
- Khan Academy**,** Microeconomics & Finance Basics**,** https://www.khanacademy.org/economics-finance-domain
- SWAYAM Portal**,** Business Economics (UGC-approved), https://swayam.gov.in

AI TOOLS USED IN THIS COURSE:

- ChatGPT / Claude AI, Concept clarification, case study generation, problem solving assistance
- Microsoft Copilot, Financial analysis, spread sheet automation, data interpretation
- Google Gemini, Research support, market structure comparisons, summarisation
- Tally AI / Zoho Books AI, Digital accounting simulations, ledger preparation, journal entries
- Statista / Tableau AI, Data visualisation for demand trends and financial ratios
- Wolfram Alpha, Quantitative problems – NPV, IRR, Break-Even, elasticity calculations
- Grammarly AI, Report writing, case study documentation, assignments

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26HS0825) UNIVERSAL HUMAN VALUES, UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT

(Common to all branches)

Prerequisite: None

Course Objectives:

The objectives of the course is

1. *Introduce the need for value education and the crisis of values in modern society through self-exploration.*
2. *Develop understanding of harmony within the individual through the Self-Body relationship.*
3. *Build awareness of harmony in the family and society through trust and the comprehensive human goal.*
4. *Explore harmony in nature and existence through the four orders and their interconnectedness.*
5. *Apply ethical human conduct principles to professional engineering practice and AI ethics.*

UNIT-I	Introduction to Value Education and Human Consciousness
Content	**Need for Value Education:** Crisis of values in modern society; Human Being: A co-existence of Self (Jeevan) and Body; Consciousness as a distinguishing feature. **Natural Acceptance:** The innate human capacity to know what is right; **Activities of the Self:** Knowing, Assuming, Recognising, Fulfiling. Difference between Animal Consciousness and Human Consciousness.
AI Integration	**AI Tool:** ChatGPT to present ethical dilemmas and compare AI responses with naturally accepted human values. Reflection Activity: Day One or Notion AI journalling assistant for natural acceptance documentation. **Discussion:** Can AI ever have human values?

UNIT-II	Harmony in the Individual
Content	**Harmony between Self and Body:** Needs of the Self (happiness, trust, etc.) vs. Needs of the Body (food, shelter, clothing); Four Dimensions: Icchha (desire), Vichar (thought), Anubhav (experience), Bodh (realisation). **Feelings in Relationships:** Nine feelings , Trust, Respect, Affection, Care, Guidance, Reverence, Glory, Gratitude, Love; Understanding Happiness and Prosperity. **Self-regulation vs. External Regulation:** Inner motivation and its role in ethical conduct.

AI Integration	**AI Tool:** Replika (free AI companion) for exploring emotional needs conversations and reflecting on AI limitations. Activity: Well-being Wheel using Canva; AI-assisted gap identification between material and value-based happiness.

UNIT-III	**Harmony in the Family and Society**
Content	**Family as the Basic Unit of Living:** Roles and responsibilities in family; Trust as the Foundational Value in Relationships; From Family to Society: Comprehensive Human Goal. **Universal Human Order:** Five dimensions of human existence (Individual, Family, Society, Nature, Existence); Social Justice: Ensuring rights, duties, and dignified participation.
AI Integration	**AI Tool:** ChatGPT to generate family conflict scenarios analysed through trust and respect. **Case Study:** AI-assisted mapping of historical social movements onto the five-order framework. Project: community service observation documentation.

UNIT-IV	**Harmony in Nature and Existence**
Content	Four Orders of Nature: Material, Plant/Bio, Animal, Human; Interconnectedness: Mutual fulfilment among the four orders; Ecological Balance: Current ecological crises and human responsibility. Sustainable Development Goals (SDGs) and Value-based Engineering; Right Understanding of Existence: Co-existence of units in space.
AI Integration	**AI Tool:** Global Forest Watch for deforestation data analysis correlated with ecological values. **AI Tool:** ChatGPT to generate sustainable engineering project ideas critiqued via ecological harmony principles. Activity: campus ecological audit using Google Earth.

UNIT-V	**Ethical Human Conduct and Professional Ethics**
Content	Right Understanding, Right Feeling, Right Action: The value-ethics-skill triad; Professional Ethics: Codes of conduct for engineers (ASME, IEEE, IEI). Corruption, Dishonesty, and Short-term Thinking: Causes and value-based remedies; Vision for a Value-based Society; AI Ethics: Bias, fairness, transparency, and accountability.
AI Integration	**AI Tool:** IBM AI Fairness 360 to demonstrate AI bias using sample datasets. **Case Study:** Bhopal Gas Tragedy Challenger Disaster through professional ethics lens. **Capstone:** Personal Value Charter drafted with ChatGPT assistance.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and understand the concept of universal human values, natural acceptance, and the purpose of human life as articulated in Indian value philosophy.	**L1,L2**
CO2	Apply the framework of values-based living to personal relationships, family harmony, and societal well-being in daily life contexts.	**L3 (Apply)**
CO3	Analyse ethical dilemmas in engineering practice using the lens of universal human values and differentiate value-driven from interest-driven conduct.	**L4 (Analyse)**
CO4	Evaluate current engineering and social practices for their alignment with sustainability, ecological balance, and universal human order.	**L5 (Evaluate)**
CO5	Create a personal value charter and an action plan for ethical engineering practice using AI reflection tools and collaborative dialogue.	**L6 (Create)**

Bloom's Taxonomy Coverage , Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1/L2	Remember/Understand	**CO1**	Natural acceptance concept recall; human consciousness identification exercises.
L3	Apply	**CO2**	Values-based living application to personal relationship scenarios.
L4	Analyse	**CO3**	Engineering ethical dilemma analysis using the universal human values lens.
L5/L6	Evaluate/Create	**CO4, CO5**	Sustainability practice evaluation; personal value charter creation.

CO , PO PSO Articulation Matrix (NBA)

Scale: 3 , Strong, 2 , Moderate, 1 , Slight, , No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	,	,	,	,	,	**2**	**2**	**3**	1	**2**	,	**2**
CO2	,	,	,	,	,	**3**	**2**	**3**	**2**	**2**	1	**2**
CO3	1	**2**	,	1	,	**3**	**2**	**3**	**2**	**2**	1	**3**
CO4	1	**2**	**2**	**2**	**2**	**3**	**3**	**3**	**2**	**2**	**2**	**3**
CO5	1	**2**	**2**	1	**2**	**3**	**2**	**3**	**3**	**3**	**2**	**3**

CO-PO Correlation Highlights	CO4 and CO5 strongly align with PO6 (Engineer & Society), PO7 (Environment & Sustainability), and PO8 (Ethics) through evaluation of sustainable practices and ethical engineering action plans. PO8 (Ethics) shows the strongest overall correlation across all COs, reflecting the course's core focus on universal human values and professional conduct.

Textbooks :

1. Gaur, R. R., Sangal, R., and Bagaria, G. P., *A Foundation Course in Human Values and Professional Ethics*, 3rd Edition, Excel Books, 2019.
2. Nagraj, A., *Human Values and Professional Ethics: Universal Human Order*, Jeevan Vidya Prakashan, 2015.

Reference Books :

1. Subramanian, C. V., *Professional Ethics and Human Values*, Oxford University Press, 2020.
2. Bajpai, S. C., *Engineering Ethics: Concepts and Cases*, McGraw-Hill Education, 2019.
3. Flood, R. L., *Rethinking the Fifth Discipline: Learning within the Unknowable*, Routledge, 2018.
4. Covey, S. R., *The 7 Habits of Highly Effective People*, 30th Anniversary Edition, Simon & Schuster, 2020.
5. Singer, P., *Practical Ethics*, 4th Edition, Cambridge University Press, 2021.

Online Resources :

1. AICTE , Value Education Resources: aicte-india.org
2. UNESCO , Ethics of AI: www.unesco.org/en/artificial-intelligence/recommendation-ethics
3. IEEE Ethics Resources: ethics.iit.edu
4. NPTEL , Human Values & Professional Ethics: swayam.gov.in
5. Jeevan Vidya Andolan: www.jeevanvidya.info

YouTube Video Resources :

1. YouTube: NPTEL , Human Values & Professional Ethics , www.youtube.com/c/nptel
2. YouTube: TED-Ed , Ethics and Moral Philosophy , www.youtube.com/@TED-Ed
3. YouTube: Michael Sandel , Justice: What's the Right Thing to Do? , search on YouTube
4. YouTube:AI Ethics Overview , IBM Technology , www.youtube.com/@IBMTechnology
5. YouTube: Crash Course , Philosophy , www.youtube.com/@crashcourse

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech - II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS3103) INTELLIGENT DATA PROCESSING AND SYSTEM INTEGRATION (CAI)

Prerequisites: Python Programming; Data Structures; Databases (DBMS); Operating Systems

Course Objectives

- Introduce data ingestion patterns — batch, streaming, ETL/ELT — and system integration techniques using REST APIs, Kafka, and integration design patterns.
- Develop competency in intelligent data processing using Pandas, NumPy, and Apache Spark for scalable, production-grade AI data pipelines.
- Build knowledge of database integration for AI systems — SQL/NoSQL from Python, ORM, vector databases, and data access patterns for AI APIs.
- Introduce REST API design and microservice integration using FastAPI and gRPC for building AI-powered services.
- Develop understanding of workflow orchestration with Apache Airflow, pipeline monitoring, error handling, and responsible data integration under GDPR and India DPDP Act 2023.

Course Content

UNIT – I	DATA INGESTION AND SYSTEM INTEGRATION PATTERNS	9 Hrs

Content	□ Data ingestion patterns: batch ingestion (scheduled, full-load vs. incremental, change data capture — CDC using Debezium); streaming ingestion (event-driven, real-time); micro-batch (Spark Structured Streaming); comparing latency, throughput, and complexity of each pattern. □ ETL vs. ELT: Extract-Transform-Load (transform before loading — traditional data warehouse); Extract-Load-Transform (load raw, transform in-place — modern cloud data stack using dbt); when to use each; EL+T hybrid pattern for AI feature pipelines. □ System integration techniques: REST API integration (requests library — GET, POST, PUT, DELETE, authentication, pagination, retry with backoff); webhook-based integration; file-based integration (SFTP, S3-compatible object storage); message queue integration (Apache Kafka — producers and consumers from Python using kafka-python). □ Integration patterns: adapter pattern (wrap external API in a standard internal interface); façade pattern (simplify a complex subsystem); event-driven integration (publish events on state changes, subscribe to external events); strangler fig pattern (incremental replacement of legacy systems with AI-powered components). **Source:** TB1: Chapters 3, 10–11 (Kleppmann — data models, batch processing, stream processing); TB3: Chapters 3–5 (Huyen — data engineering, feature pipelines)

AI Integration	Use ChatGPT to explain CDC vs. full-load batch ingestion with a concrete e-commerce database example; use Claude to design an integration pattern for connecting a legacy ERP system to an AI prediction service; use an AI assistant to generate retry-with-backoff code for a REST API integration.

UNIT – II	**INTELLIGENT DATA PROCESSING**	**9 Hrs**

Content	□ NumPy for intelligent processing: vectorised operations vs. loops (performance comparison); broadcasting for batch computations; linear algebra (dot, matmul, linalg.solve) for AI feature computation; array manipulation (reshape, stack, split, where, argsort); memory-efficient dtypes (float32 vs. float64 for AI workloads). □ Pandas for intelligent data processing: method chaining with pipe(); assign() for derived columns; query() for readable filtering; apply() with custom functions; groupby with transform() for group-level enrichment without reducing rows; merge_asof() for time-series joins. □ Data quality in processing pipelines: inline validation using assert statements and custom functions; detecting and handling schema drift; data freshness checks; building a processing audit log (input row count, output row count, null rate before/after, processing time). □ Apache Spark for distributed processing: SparkSession, DataFrames, lazy evaluation DAG; transformations (select, filter, withColumn, groupBy, join, window functions) vs. actions (show, count, collect, write); Spark SQL; UDFs for custom AI logic; reading from and writing to Parquet, Delta Lake, and JDBC. **Source:** TB2: Chapters 4–12 (McKinney — NumPy, Pandas, groupby, time-series); TB1: Chapter 10 (Kleppmann — batch processing with MapReduce and Spark)
AI Integration	Use ChatGPT to explain Spark lazy evaluation with a DAG example; use Claude to identify performance bottlenecks in a given Pandas processing pipeline and suggest vectorised alternatives; use GitHub Copilot to implement a Spark UDF for a custom text processing step.

UNIT – III	**DATABASE INTEGRATION FOR AI SYSTEMS**	**9 Hrs**

Content	□ SQL database integration with Python: SQLAlchemy Core (connection, execute, text()) and ORM (declarative base, session, model classes, relationships); connection pooling — QueuePool, NullPool; bulk insert (bulk_insert_mappings), upsert (on_conflict_do_update for PostgreSQL); Alembic for database migrations. □ NoSQL integration: pymongo — connecting to MongoDB, CRUD operations, aggregation pipeline from Python, indexing; Redis integration with redis-py — key-value storage, TTL, sorted sets; Cassandra with

	cassandra-driver — session, prepared statements, batch operations; choosing NoSQL for AI: document store for ML experiment metadata, key-value for feature serving, column-family for time-series sensor data. • Vector database integration: FAISS from Python — IndexFlatL2, IndexIVFFlat, index serialisation; Chroma — PersistentClient, collection.add(), collection.query(); Pinecone client — upsert, query with metadata filter; pattern: embed text/image → store vector+metadata → query by similarity; hybrid search (vector + scalar filter). • Data access patterns for AI systems: repository pattern (abstract data access behind an interface); CQRS (Command Query Responsibility Segregation — separate read and write models for high-throughput AI APIs); read-through and write-through caching with Redis; connection pool tuning for concurrent AI inference requests. **Source:** TB4: Chapters 5–6, 10 (Elmasri & Navathe — SQL, NoSQL, advanced data management); TB3: Chapter 5 (Huyen — feature stores, data access for ML)
AI Integration	Use ChatGPT to explain the SQLAlchemy ORM session lifecycle; use Claude to design a repository pattern for an AI application that needs to switch between SQLite (development) and PostgreSQL (production); use an AI assistant to explain FAISS index types and when to use each.

UNIT – IV	**API DESIGN AND MICROSERVICE INTEGRATION**	**9 Hrs**

Content	• REST API design principles: resource naming conventions, HTTP methods (GET, POST, PUT, PATCH, DELETE) and their semantics, status codes (2xx, 4xx, 5xx), idempotency, statelessness; versioning strategies (URL versioning /v1/, header versioning); pagination (cursor-based vs. offset-based); OpenAPI specification (Swagger) for API documentation. • FastAPI for AI system integration: path operations, request/response models with Pydantic, path and query parameters, request body validation; dependency injection for database sessions and model loading; background tasks for async processing; middleware for logging, CORS, and rate limiting; lifespan events for model warm-up on startup. • Asynchronous processing and inter-service communication: async/await in Python for non-blocking I/O; asyncio event loop; aiohttp for async HTTP clients; gRPC from Python — protobuf schema definition, code generation, unary and server-streaming RPCs; gRPC vs. REST for internal AI microservice communication. • AI API integration patterns: model-as-a-service (expose ML model via REST API); circuit breaker pattern (fail fast when downstream service is unavailable using tenacity or circuitbreaker library); API gateway pattern (single entry point for multiple AI services — authentication, rate limiting, routing); webhook callbacks for async AI job completion. **Source:** TB3: Chapters 7, 9 (Huyen — serving infrastructure, model

	deployment); TB1: Chapter 12 (Kleppmann — the future of data systems, integration)
AI Integration	Use ChatGPT to design a FastAPI endpoint for a text classification model with input validation and background logging; use Claude to explain the circuit breaker pattern with a concrete AI microservice failure scenario; use GitHub Copilot to generate a gRPCprotobuf schema for a model prediction service.

UNIT – V	**WORKFLOW ORCHESTRATION AND RESPONSIBLE INTEGRATION**	**9 Hrs**

Content	□ Apache Airflow for workflow orchestration: DAG (Directed Acyclic Graph) structure — tasks, operators, dependencies; core operators — PythonOperator, BashOperator, BranchPythonOperator, ShortCircuitOperator; scheduling (cron expressions, timedelta); XCom for inter-task communication; task retries, retry_delay, timeout; Airflow connections and variables for credential management. □ Pipeline monitoring and observability: structured logging (Python logging module with JSON formatter); metrics collection (Prometheus client library — counters, gauges, histograms); distributed tracing (OpenTelemetry — trace IDs across service calls); alerting on pipeline failures (Airflow email/Slack callbacks, Prometheus Alertmanager); data quality monitoring (Great Expectations checkpoints in Airflow DAGs). □ Error handling and resilience in integrated systems: exception hierarchy design for pipeline errors (DataIngestionError, TransformationError, IntegrationError); dead-letter queues (DLQ) for unprocessable messages; idempotent processing (safe to re-run without side effects); saga pattern for distributed transactions across multiple databases/services; graceful degradation (return cached result when live inference fails). □ Responsible system integration: PII detection in data pipelines (using spaCy NER or regex to flag names, emails, phone numbers, Aadhaar, PAN before persistence); data minimisation in integration design; consent-aware data routing (route data tagged with consent flags to appropriate storage); GDPR Article 5 and India DPDP Act 2023 applied to integrated AI systems — data subject rights (access, erasure) in a multi-database architecture; audit logging for compliance. **Source:** TB1: Chapters 9–12 (Kleppmann — consistency, reliability, batch and stream processing integration); TB3: Chapter 11 (Huyen — responsible AI, privacy)
AI Integration	Use ChatGPT to explain Airflow XCom with a concrete multi-task DAG example; use Claude to design a responsible data routing pipeline that respects consent flags under India DPDP Act 2023; use an AI assistant to generate a structured logging configuration with JSON formatter for a FastAPI application.

Course Outcomes (COs

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Explain batch and streaming data ingestion patterns; design ETL/ELT pipelines using connectors, APIs, and message queues. (L4)	L4 (Analyse)
CO2	Apply Pandas and NumPy for intelligent data processing; implement Spark DataFrames for distributed transformation. (L3)	L3 (Apply)
CO3	Integrate SQL, NoSQL, and vector databases with Python applications using SQLAlchemy, pymongo, and FAISS. (L3)	L3 (Apply)
CO4	Design and implement REST APIs and gRPC services using FastAPI for AI system integration. (L4)	L4 (Analyse)
CO5	Orchestrate multi-step data workflows using Apache Airflow; apply logging, monitoring, and GDPR/DPDP compliance to integrated AI systems. (L5)	L5 (Evaluate)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1, CO3**	Define CDC, ETL/ELT, SQLAlchemy ORM, FAISS index types, DAG, XCom, circuit breaker; describe Kafka producer-consumer model.
L3	Apply	**CO2, CO3**	Apply Pandas method chaining; implement SQLAlchemy ORM CRUD; query FAISS vector index; build Spark DataFrame transformations.
L4	Analyse	**CO1, CO4**	Design REST API with FastAPI; analyse CDC vs. full-load trade-offs; compare gRPC vs. REST for AI microservices; trace Airflow DAG execution.
L5	Evaluate	**CO5**	Evaluate pipeline monitoring signals; assess GDPR and DPDP Act 2023 compliance checklist; evaluate circuit breaker vs. retry strategies.
L6	Create	**CO4, CO5**	Design a complete FastAPI + Airflow AI data integration system with observability, error resilience, and GDPR compliance.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgm
CO1	L4	**3**	**3**	**3**	**2**	**2**	1	–	–	–	**2**	1
CO2	L3	**3**	**3**	**2**	**2**	**3**	–	–	–	1	1	–
CO3	L3	**3**	**3**	**2**	**2**	**3**	–	–	–	1	1	–
CO4	L4	**3**	**3**	**3**	**2**	**3**	1	–	–	1	**2**	1
CO5	L5	**3**	**3**	**3**	**3**	**3**	**2**	1	**2**	1	**2**	1

CO-PO Correlation Highlights	Strong PO5 (Modern Tools) through FastAPI, Airflow, SQLAlchemy, pymongo, FAISS, Chroma, kafka-python, Spark, Prometheus, and OpenTelemetry. PO3 (Design/Development) through API design, microservice integration patterns, and Airflow DAG orchestration. PO6 (Engineer & Society) and PO8 (Ethics) through PII detection, consent-aware data routing, and GDPR/India DPDP Act 2023 data subject rights implementation in Unit V.

LEARNING RESOURCES

Textbooks

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017. (Primary — Units I, II, V)
2. Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022. (Unit II — NumPy, Pandas)
3. Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022. (Units I, III, IV, V)
4. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson. (Unit III)

Reference Books

1. Sam Newman, Building Microservices, 2nd Edition, O'Reilly Media, 2021.
2. Bas Geerdink, A Practical Guide to Building Ethical AI Systems, Manning, 2023.

Online Resources

- Apache Airflow – airflow.apache.org |FastAPI – fastapi.tiangolo.com
- SQLAlchemy – sqlalchemy.org | Apache Spark – spark.apache.org
- FAISS – github.com/facebookresearch/faiss | Chroma – trychroma.com
- Great Expectations – greatexpectations.io |OpenTelemetry Python – opentelemetry.io
- NPTEL – Data Science for Engineers, IIT Madras – nptel.ac.in
- India DPDP Act 2023 – meity.gov.in | GDPR – gdpr.eu

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0512) INTELLIGENT DATABASE MANAGEMENT SYSTEMS
(Professional Core -V)
(Common to CSE & Allied branches)

Course Objectives :

The objectives of the course is to

1. *Understand fundamental concepts of database systems, data models, and DBMS architecture.*
2. *Design relational databases using ER modelling and normalization techniques.*
3. *Master SQL for data definition, manipulation, querying, and transaction control.*
4. *Explore transaction management, concurrency control, and database recovery.*
5. *Introduce modern database systems (NoSQL, vector databases) and AI-assisted data handling.*

UNIT – I	INTRODUCTION TO DBMS AND DATA MODELLING	9 Hrs

Content	□ File system vs. DBMS: data redundancy, inconsistency, advantages of DBMS; 3-level DBMS architecture (internal, conceptual, external); logical and physical data independence; data models — hierarchical, network, relational, object-oriented. □ ER Model: entities (strong and weak), attributes (simple, composite, multivalued, derived), relationships; keys (primary, candidate, composite); cardinality ratios (1:1, 1:N, M:N) and participation constraints (total, partial); ER diagrams. □ ER-to-relational mapping: rules for entities, weak entities, binary relationships, multivalued attributes; introduction to centralized vs. distributed and cloud database systems; overview of NoSQL categories. **Source:** TB1: Chapters 1–3, 7–9 (Silberschatz, Korth & Sudarshan, 7th ed.); TB2: Chapters 1–3, 7–9 (Elmasri & Navathe, 7th ed.)
AI Integration	Use draw.io and MySQL Workbench to design ER diagrams; use ChatGPT to convert natural language problem statements into ER models and validate the output manually.

UNIT – II	RELATIONAL MODEL AND SQL	9 Hrs

Content	□ Relational model: relations, tuples, attributes, domains, schemas; relational algebra — selection (σ), projection (π), union, set difference, Cartesian product, natural join, outer joins, intersection.

	□ SQL: DDL (CREATE, ALTER, DROP); DML (INSERT, UPDATE, DELETE); DQL (SELECT with WHERE, ORDER BY, GROUP BY, HAVING, aggregate functions, joins, subqueries, set operations); DCL (GRANT, REVOKE); TCL (COMMIT, ROLLBACK, SAVEPOINT). □ Views, indexes, and query optimization: simple and complex views; B-Tree and B+ Tree indexing — dense vs. sparse; primary and secondary indexes; query optimization basics — cost-based optimizer, predicate pushdown. **Source:** TB1: Chapters 2–6, 11 (Silberschatz et al.); TB2: Chapters 5–6, 14–15 (Elmasri & Navathe)
AI Integration	Use MySQL and SQLite for SQL query implementation; use ChatGPT to generate SQL from natural language and verify manually; use GitHub Copilot to scaffold complex queries; use Google Colab for SQL exercises.

UNIT – III	**DATABASE DESIGN AND NORMALIZATION**	**9 Hrs**

Content	□ Functional dependencies (FDs): trivial and non-trivial FDs; Armstrong's axioms; closure of FD sets; canonical (minimal) cover; types of dependencies — partial, transitive, multivalued. □ Normalization: 1NF (atomic values, no repeating groups), 2NF (no partial dependency), 3NF (no transitive dependency), BCNF (every determinant is a superkey); introduction to 4NF and 5NF. □ Decomposition: lossless join decomposition; dependency preservation; BCNF decomposition algorithm; denormalization trade-offs for OLAP workloads. **Source:** TB1: Chapters 7–8 (Silberschatz et al.); TB2: Chapters 14–15 (Elmasri & Navathe)
AI Integration	Use ChatGPT to input a schema and get normalized tables; validate normalization steps manually; use Jupyter Notebook for FD closure calculations; practice AI-assisted schema design and verify with Armstrong's axioms.

UNIT – IV	**TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL**	**9 Hrs**

Content	□ Transaction concepts: ACID properties (Atomicity, Consistency, Isolation, Durability); transaction states; COMMIT, ROLLBACK, SAVEPOINT; isolation levels — READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE. □ Concurrency control: lock-based protocols — shared and exclusive locks, two-phase locking (2PL), strict 2PL; timestamp-based protocols; deadlock detection (waits-for graph), prevention (wound-wait, wait-die), avoidance. □ Serializability: conflict serializability; precedence graph construction; view serializability. Recovery: log-based recovery — undo and redo; Write-Ahead Log (WAL); checkpointing; ARIES algorithm (Analysis, Redo, Undo). **Source:** TB1: Chapters 14–17 (Silberschatz et al.); TB2: Chapters 19–21 (Elmasri & Navathe)

AI Integration	Use Python simulations to create concurrent transaction schedules; use ChatGPT to analyse serializability and trace precedence graph construction; simulate lost update and dirty read scenarios and resolve.

UNIT – V	**INTELLIGENT DATA MANAGEMENT FOR AI SYSTEMS**	**9 Hrs**

Content	□ NoSQL systems: motivation (3Vs, schema flexibility, horizontal scale); CAP theorem; document stores (MongoDB — aggregation pipeline), key-value (Redis — TTL, sorted sets), column-family (Cassandra — partition key), graph (Neo4j — Cypher). □ Text-to-SQL and LLM-over-database: natural language to SQL translation; schema-linking; accuracy challenges; RAG-over-SQL (retrieve rows → LLM context → answer); LLM as data analyst; verification requirements before production use. □ Vector data management: vector databases (FAISS, Pinecone, Weaviate, Chroma); ANN search (HNSW, IVF); similarity metrics (cosine, Euclidean, dot product); hybrid search (vector + scalar filter); re-ranking. Data governance: data catalog (Apache Atlas), lineage tracking; GDPR Article 5; India DPDP Act 2023. **Source:** TB1: Chapters 10, 22–23 (Silberschatz et al.); TB2: Chapter 24–25 (Elmasri & Navathe); Kleppmann, Designing Data-Intensive Applications (O'Reilly)
AI Integration	Use ChatGPT to generate SQL from natural language and verify each query; use Claude to compare HNSW vs. IVF for a vector search workload; use MongoDB Atlas for document store experiments; use an AI assistant to generate a data governance checklist.

Course Outcomes

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

CO	Course Outcome Statement	Bloom's Level
CO1	Explain DBMS architecture, data models, and ER modelling; design ER diagrams and map them to relational schemas.	L4 (Analyze)
CO2	Apply relational algebra and write SQL queries for data definition, manipulation, retrieval, and control.	L4 (Analyze)
CO3	Apply normalization techniques (1NF through BCNF) to eliminate redundancy and design efficient relational schemas.	L4 (Analyze)
CO4	Analyse transaction schedules for serializability; apply concurrency control protocols and recovery mechanisms.	L4 (Analyze)

CO	Course Outcome Statement	Bloom's Level
CO5	Evaluate modern database systems including NoSQL and vector databases; apply AI-assisted querying and data governance.	L5 (Evaluate)

CO – PO / PSO Articulation Matrix (NBA)

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO1	L4	**3**	**3**	**3**	**2**	1	1	–	–	–	1	1
CO2	L4	**3**	**3**	**2**	**2**	**2**	–	–	–	–	1	–
CO3	L4	**3**	**3**	**3**	**2**	1	–	–	–	–	1	–
CO4	L4	**3**	**3**	**2**	**3**	1	–	–	–	–	1	–
CO5	L5	**3**	**3**	**3**	**3**	**3**	1	–	–	1	**2**	1

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO-PO Correlation Highlights	Strong PO1 and PO2 through DBMS theory, relational algebra, and schema design. PO3 through ER-to-schema mapping, normalisation, and AI-enabled query design. PO4 (Investigation) through transaction schedule analysis and concurrency control evaluation. PO5 through MySQL, MongoDB, FAISS, Chroma, and ChatGPT-assisted querying.

Text Books :

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, 7th Edition, McGraw-Hill, 2019.
2. Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, 7th Edition, Pearson Education, 2016.

Reference Books :

1. Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill.
2. Martin Kleppmann, *Designing Data-Intensive Applications*, O'Reilly Media, 2017.

Online resources :

1. NPTEL – Database Management Systems, IIT Madras – nptel.ac.in
2. MySQL – dev.mysql.com/doc | MongoDB – docs.mongodb.com | Stanford – online.stanford.edu
3. India DPDP Act 2023 – meity.gov.in|FAISS– github.com/facebookresearch/faiss

Recommended Free &Free-Tier Ai/Database Tools

1. **Design & Modeling:** draw.io; MySQL Workbench – ER diagrams and schema design
2. **SQL Databases:** MySQL; SQLite (free, open-source) – query implementation and practice
3. **NoSQL & Vector:** MongoDB Atlas; Redis; Neo4j; FAISS; Chroma – modern data stores
4. **Environment:** Jupyter Notebook / Google Colab (free) – FD closure, query exercises
5. **AI Assistance:** ChatGPT; Claude; GitHub Copilot – text-to-SQL, normalization, and verification

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech - II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS3104) COMPUTER ARCHITECTURE FOR INTELLIGENT SYSTEMS

Prerequisites: Digital Logic Design; Python Programming; Data Structures; Operating Systems (basic)

Course Objectives

- Introduce computer organization fundamentals — ISA, RISC-V, data representation — and explain how they shape AI software performance.
- Develop understanding of CPU microarchitecture — pipelining, hazards, out-of-order execution, and SIMD — as the foundation for AI workload throughput.
- Build knowledge of memory systems — cache hierarchy, virtual memory, NUMA — and explain the memory wall's impact on AI workloads.
- Introduce I/O subsystems, storage hierarchy, and PCIe interconnects as the infrastructure connecting CPUs to AI accelerators.
- Develop understanding of OS-level mechanisms — scheduling, NUMA-aware allocation, containers, power management — for AI workload performance optimisation.

Course Content

UNIT – I	COMPUTER ORGANIZATION AND ISA	9 Hrs

Content	□ Computer organization: Von Neumann architecture — fetch-decode-execute cycle, registers, ALU, control unit, memory bus; Harvard architecture — separate instruction and data memories (used in microcontrollers and edge AI chips); stored-program concept; performance equation: CPU time = instruction count × CPI × clock cycle time. □ Number representation for AI: signed integers (two's complement); IEEE 754 floating-point — float32 (23-bit mantissa), float16 (10-bit mantissa, 5-bit exponent), bfloat16 (8-bit mantissa, 8-bit exponent — used in TPUs and modern GPUs for AI training); precision trade-offs — float32 for training, float16/bfloat16 for inference; overflow, underflow, NaN, infinity handling. □ Instruction Set Architecture (ISA): ISA as the hardware-software interface; CISC (x86 — complex variable-length instructions, memory-to-memory operations) vs. RISC (ARM, RISC-V — fixed-length, load-store, simple instructions); RISC-V ISA — base integer ISA (RV32I, RV64I), extensions (M: multiply/divide, F: single-precision float, D: double-precision, V: vector for AI workloads); RISC-V instruction formats (R, I, S, B, U, J). □ Amdahl's Law and performance analysis: Amdahl's Law — speedup limited by the non-parallelizable fraction; application to AI: if 90% of a training loop is matrix multiply and we accelerate it 100×, total speedup ≤ 10×; strong scaling vs. weak scaling; FLOPS as an AI performance metric; TFLOPS of modern AI hardware.

	Source: TB1: Chapters 1–3 (Patterson & Hennessy, CO&D RISC-V — performance, RISC-V ISA, number representation); TB2: Chapter 1 (Hennessy & Patterson, CA:AQA — fundamentals of quantitative design)
AI Integration	Use ChatGPT to derive Amdahl's Law speedup for a given AI training loop breakdown; use Claude to explain bfloat16 vs. float16 precision trade-offs for a transformer training scenario; use an AI assistant to decode a RISC-V instruction encoding in binary.

UNIT – II	**CPU MICROARCHITECTURE FOR AI WORKLOADS**	**9 Hrs**

Content	□ Pipelining: five-stage RISC-V pipeline (IF, ID, EX, MEM, WB); pipeline CPI = 1 ideally; structural hazards (resource conflicts — solution: stall or duplicate hardware); data hazards — RAW (Read After Write) — forwarding/bypassing paths; WAW and WAR hazards; control hazards — branch prediction (static: predict not-taken; dynamic: 2-bit saturating counter, branch history table, branch target buffer). □ Out-of-order execution and superscalar: in-order vs. out-of-order execution; Tomasulo's algorithm — reservation stations, common data bus (CDB), register renaming to eliminate WAW/WAR hazards; reorder buffer (ROB) for precise exceptions; superscalar execution — issuing multiple instructions per cycle; issue width and IPC; modern x86 and ARM cores (4–8 wide issue). □ SIMD and vector processing for AI: SIMD (Single Instruction Multiple Data) — same operation on multiple data elements simultaneously; x86 SIMD extensions — MMX (64-bit), SSE/SSE2 (128-bit), AVX/AVX2 (256-bit), AVX-512 (512-bit); Python NumPy and PyTorch automatically use AVX-512 via optimized BLAS libraries; SIMD for matrix multiply: vectorised dot product of 16 float32 values in one AVX-512 instruction; RISC-V V extension. □ CPU performance analysis for AI: CPI breakdown — compute-bound vs. memory-bound operations; roofline model — arithmetic intensity (FLOP/byte), compute ceiling (peak FLOPS), bandwidth ceiling (peak GB/s × arithmetic intensity); identifying whether a given AI layer (attention, convolution, fully-connected) is compute-bound or memory-bound; Intel VTune and perf tool concepts for CPU profiling. **Source:** TB1: Chapters 4–5 (Patterson & Hennessy — pipelining, data hazards, control hazards); TB2: Chapters 3–4 (Hennessy & Patterson — ILP, out-of-order, SIMD)
AI Integration	Use ChatGPT to trace Tomasulo's algorithm for a 5-instruction sequence with RAW hazards; use Claude to apply the roofline model to a transformer self-attention layer — determine if it is compute-bound or memory-bound on a given CPU; use an AI assistant to explain why NumPy matrix multiply is faster than a Python loop (BLAS + SIMD).

UNIT – III	MEMORY SYSTEMS AND THE MEMORY WALL	9 Hrs

Content	□ Cache hierarchy: SRAM cache vs. DRAM main memory — speed, cost, capacity trade-offs; direct-mapped, set-associative (2-way, 4-way, 8-way), and fully-associative caches; cache parameters — cache size, block size, associativity; cache miss types — compulsory (cold), capacity, conflict; replacement policies — LRU, pseudo-LRU, random; write policies — write-through vs. write-back with write-allocate. □ Cache performance and locality: average memory access time (AMAT = hit time + miss rate × miss penalty); multilevel cache (L1, L2, L3) AMAT formula; spatial locality (accessing nearby memory) and temporal locality (reusing same data); cache-oblivious algorithms; how cache blocking (loop tiling) improves AI training performance. □ Virtual memory and memory management: virtual address space; page table; TLB (Translation Lookaside Buffer) — TLB hit vs. miss; page fault; page replacement policies (LRU, clock algorithm); huge pages (2 MB vs. 4 KB) — reducing TLB pressure for large ML model weight tensors; memory-mapped files (mmap) for large dataset loading used by PyTorchDataLoader. □ DRAM architecture and the memory wall: DRAM cell, row/column addressing, RAS/CAS timing; DRAM bandwidth (GB/s) vs. DRAM latency (ns); DDR4/DDR5 specifications; NUMA (Non-Uniform Memory Access) — multi-socket systems, local vs. remote DRAM access time; memory bandwidth as the bottleneck for LLM inference (KV-cache is memory-bound); roofline analysis: transformer inference is memory-bound, training is compute-bound. **Source:** TB1: Chapter 5 (Patterson & Hennessy — large and fast: exploiting memory hierarchy); TB2: Chapter 2 (Hennessy & Patterson — memory hierarchy design, DRAM, NUMA)
AI Integration	Use ChatGPT to compute AMAT for a given two-level cache hierarchy with miss rates; use Claude to explain how loop tiling improves matrix multiply cache performance with a concrete 4×4 tile example; use an AI assistant to explain why LLM inference is memory-bound using the roofline model for a given model size and hardware bandwidth.

UNIT – IV	I/O, STORAGE, AND INTERCONNECTS FOR AI SYSTEMS	9 Hrs

Content	□ I/O subsystem: polling vs. interrupt-driven I/O vs. DMA (Direct Memory Access) — DMA controller transfers data between I/O device and DRAM without CPU involvement; IOMMU (I/O Memory Management Unit) for secure DMA from accelerators; PCIe generations (PCIe 3.0: 8 GB/s, PCIe 4.0: 16 GB/s, PCIe 5.0: 32 GB/s per ×16 slot) and their impact on GPU-to-CPU data transfer bandwidth. □ Storage hierarchy for AI systems: register → L1/L2/L3 cache → DRAM → NVMe SSD → SATA SSD → HDD → network storage (S3, NFS); latency

	and bandwidth at each level; NVMe SSD (PCIe 4.0 ×4 — up to 7 GB/s sequential read, ~100 µs latency) vs. SATA SSD (600 MB/s) vs. HDD (150 MB/s, 10 ms seek); GPUDirect Storage — data path from NVMe directly to GPU VRAM bypassing CPU DRAM. • Hardware accelerator attachment model: PCIe device enumeration and BAR (Base Address Register) mapping; CPU-GPU memory model — host (DRAM) vs. device (VRAM) memory; cudaMalloc, cudaMemcpy (H2D, D2H, D2D); unified memory (CUDA UM); pinned (page-locked) host memory — faster H2D transfer; PCIe bandwidth as the bottleneck for small batch inference. • Profiling and performance counters: hardware performance counters — CPU cycle count, cache miss count, branch misprediction count, instructions retired; Intel PMU; perf stat for Linux profiling; top-down microarchitecture analysis — frontend bound, backend bound, bad speculation, retiring; energy efficiency metrics — GFLOPS/W, performance per watt. **Source:** TB2: Chapters 1, 5 (Hennessy & Patterson — interconnects, storage, I/O performance); TB3: Chapters 4, 7 (Reddi et al. — hardware for ML, storage and I/O for AI)
AI Integration	Use ChatGPT to explain DMA data transfer for a GPU training step — trace bytes from NVMe SSD through DRAM to GPU VRAM; use Claude to explain PCIe bandwidth bottleneck for small batch GPU inference; use an AI assistant to interpret a perf stat output for a NumPy matrix multiply — identify the dominant performance bottleneck.

UNIT – V	**OPERATING SYSTEMS AND AI WORKLOAD MANAGEMENT**	**9 Hrs**

Content	• Process and thread model for AI: process (independent address space, OS-scheduled), thread (shared address space, lighter weight), coroutine/async (cooperative, no preemption — used by Python asyncio); Python Global Interpreter Lock (GIL) — prevents true CPU parallelism in CPython for pure Python code; GIL release for C-extension calls (NumPy, PyTorch); multiprocessing vs. multithreading for data loading in PyTorch (num_workers). • CPU scheduling for AI workloads: preemptive scheduling (FCFS, SJF, Round Robin, priority scheduling); real-time scheduling (EDF — Earliest Deadline First) for latency-sensitive inference; CPU affinity — pinning a process to specific cores to reduce cache invalidation (taskset); NUMA-aware scheduling — schedule threads on cores local to the NUMA node holding their data; cgroups for resource isolation of AI containers. • Memory management for AI: virtual memory recap; memory allocators — glibc malloc, jemalloc, tcmalloc (used by TensorFlow); memory fragmentation and its impact on long-running AI training jobs; huge pages (transparent huge pages — THP) for large tensor allocations; NUMA-aware allocation (numactl --membind, --cpunodebind); copy-on-write (COW) for forked DataLoader worker processes in PyTorch. • Containers and virtualization for AI: OS virtualization — hypervisors (Type

	1: VMware ESXi, KVM; Type 2: VirtualBox); containerization — cgroups + namespaces; Docker for AI model packaging; Kubernetes for AI service orchestration — Pod, Deployment, Service, resource limits (requests/limits for CPU, memory, nvidia.com/gpu); power management — CPU frequency scaling (intel_pstate governor), thermal throttling impact on sustained AI workloads. **Source:** TB4: Chapters 3–4, 9 (Silberschatz et al. — processes, threads, memory management); TB3: Chapters 9–10 (Reddi et al. — software stack for ML, deployment)
AI Integration	Use ChatGPT to explain the Python GIL and why PyTorchDataLoader workers use multiprocessing instead of multithreading; use Claude to explain NUMA-aware memory allocation for a 2-socket server running a large language model inference service; use an AI assistant to write a Dockerfile for a FastAPI-based AI inference service.

Course Outcomes (COs

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Explain ISA design principles, RISC vs. CISC trade-offs, and RISC-V instruction encoding; apply number representations for AI data types. (L2)	L2 (Understand)
CO2	Describe CPU pipelining, hazards, out-of-order execution, and SIMD/AVX; analyse how these mechanisms affect AI workload throughput. (L4)	L4 (Analyse)
CO3	Explain cache hierarchy, locality principles, virtual memory, and NUMA; identify memory bottlenecks in AI algorithms using the roofline model. (L4)	L4 (Analyse)
CO4	Describe DMA, PCIe, NVMe storage, and the hardware accelerator attachment model; explain storage hierarchy choices for ML datasets. (L2)	L2 (Understand)
CO5	Explain OS scheduling, NUMA-aware memory allocation, containerization, and power management for AI workload performance optimization. (L4)	L4 (Analyse)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1, CO4**	Define Von Neumann architecture, RISC vs. CISC, bfloat16, DMA, PCIe, NUMA, TLB; describe RISC-V instruction formats.
L3	Apply	**CO1**	Apply Amdahl's Law to a given AI training loop; convert float32 to bfloat16; decode a RISC-V binary

			instruction.
L4	Analyse	**CO2, CO3, CO5**	Trace Tomasulo's algorithm; apply roofline model to classify AI layers; compute AMAT for a multilevel cache; analyse NUMA scheduling for LLM inference.
L5	Evaluate	**CO3, CO5**	Evaluate cache blocking strategy vs. naive traversal; assess Docker resource limits for an AI inference container; evaluate pipeline CPI with hazards.
L6	Create	**CO2, CO5**	Design a CPU-optimised AI inference configuration using SIMD, cache blocking, NUMA binding, and Docker packaging.

CO – PO / PSO Articulation Matrix (NBA)

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgm
CO1	L2	**3**	**2**	1	**2**	**2**	–	–	–	–	1	–
CO2	L4	**3**	**3**	**2**	**3**	**2**	–	–	–	–	1	–
CO3	L4	**3**	**3**	**2**	**3**	**2**	–	–	–	–	1	–
CO4	L2	**3**	**2**	1	**2**	**2**	–	–	–	–	1	–
CO5	L4	**3**	**3**	**2**	**2**	**2**	–	–	–	1	1	–

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO-PO Correlation Highlights	Strong PO1 (Engineering Knowledge) through ISA, pipeline, memory hierarchy, and I/O subsystem fundamentals. PO4 (Investigation) through roofline model analysis, cache performance experiments, AMAT computation, and perf stat profiling. PO5 (Modern Tools) through RISC-V simulators, NumPy BLAS benchmarking, Linux perf, Intel VTune concepts, and Docker for AI deployment in CO5.

LEARNING RESOURCES

Textbooks

1. David A. Patterson and John L. Hennessy, Computer Organization and Design (RISC-V Edition), Morgan Kaufmann.
2. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann.
3. Vijay Janapa Reddi et al., ML Systems: Principles and Practices (free at mlsysbook.ai).
4. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th Edition, Wiley.

Reference Books

1. William Stallings, Computer Organization and Architecture, 11th Edition, Pearson Education.
2. Onur Mutlu, Computer Architecture Lectures, Carnegie Mellon University (free on YouTube).

Online Resources

- NPTEL – Computer Architecture, IIT Madras – nptel.ac.in
- MIT 6.004 Computation Structures – ocw.mit.edu
- RISC-V Specifications – riscv.org/technical/specifications
- Intel x86 Intrinsics Guide (SIMD/AVX) – software.intel.com/sites/landingpage/IntrinsicsGuide
- MLSys Book – mlsysbook.ai | Perf Wiki – perf.wiki.kernel.org

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech - II Sem.

L	T	Pr	P	C
2	0	2	0	3

(26CS3102) FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE
(Common to CSM, CAI & CAD)

Prerequisite: Data Structures; Probability and Statistics; Discrete Mathematics; Python Programming

Course Objectives

- Introduce the fundamentals, history, and modern landscape of Artificial Intelligence.
- Develop understanding of intelligent agents and search-based problem solving.
- Explore knowledge representation, logical reasoning, and classical planning.
- Provide foundational knowledge of reinforcement learning and natural language processing.
- Introduce AI applications in robotics, computer vision, and AI ethics and governance.

Course Content

UNIT – I	INTRODUCTION TO AI AND INTELLIGENT AGENTS	9 Hrs

Content	□ What is AI: definitions, foundations (philosophy, mathematics, neuroscience, linguistics, computer science); history of AI — symbolic AI, expert systems, machine learning era, deep learning, generative AI and LLMs. □ Intelligent agents: agents and environments; rationality; PEAS framework; environment properties — observable, deterministic, episodic, static, discrete, single/multi-agent. □ Agent types: simple reflex, model-based reflex, goal-based, utility-based, learning agents. **Source:** TB1: Chapters 1–2 (Russell & Norvig, AIMA 4th ed.); TB2: Chapters 1–2 (Nilsson)
AI Integration	Use ChatGPT to apply the PEAS framework to 5 real-world AI systems; use Claude to compare capabilities and limitations of GPT-4, Claude, and Gemini; use NotebookLM to generate study notes from AIMA chapters.

UNIT – II	SOLVING PROBLEMS BY SEARCHING	9 Hrs

Content	□ Problem-solving agents: state space, initial state, goal test, actions, path cost; toy problems (8-puzzle, 8-queens) and real-world problems (route finding, TSP). □ Uninformed search: BFS, uniform-cost search, DFS, depth-limited search,

	iterative deepening DFS, bidirectional search — completeness, optimality, time and space complexity. • Informed search: greedy best-first search; A* search — admissible and consistent heuristics; heuristic design from relaxed problems. • Beyond classical search: local search — hill climbing, simulated annealing, genetic algorithms; adversarial search — minimax, alpha-beta pruning; online search agents. **Source:** TB1: Chapters 3–5 (Russell & Norvig); TB2: Chapters 3–6 (Nilsson)
AI Integration	Use ChatGPT to trace BFS, DFS, and A* on a state space graph; use Claude to explain alpha-beta pruning on a game tree with node values; use VisuAlgo or an AI assistant to visualize search algorithm execution.

UNIT – III	**KNOWLEDGE REPRESENTATION AND PLANNING**	**9 Hrs**

Content	• Knowledge-based agents: the Wumpus World; propositional logic — syntax, semantics, inference, resolution, model checking (DPLL). • First-order logic (FOL): syntax — constants, predicates, functions, quantifiers; unification; forward chaining, backward chaining, resolution; knowledge engineering. • Probabilistic reasoning: uncertainty and probability; Bayes' theorem; Bayesian networks — structure, CPTs, variable elimination. • Classical planning: STRIPS representation; planning algorithms — forward and backward state-space planning; introduction to MDPs. **Source:** TB1: Chapters 7–9, 11 (Russell & Norvig); TB2: Chapters 7–11 (Nilsson)
AI Integration	Use ChatGPT to trace forward and backward chaining on a knowledge base; use Claude to explain a Bayesian network with a medical diagnosis example; use an AI assistant to explain STRIPS planning with a blocks world example.

UNIT – IV	**REINFORCEMENT LEARNING AND NATURAL LANGUAGE PROCESSING**	**9 Hrs**

Content	• Reinforcement learning: MDP formulation — states, actions, rewards, policy, value function; passive RL — TD learning; active RL — Q-learning, SARSA; policy search; applications of RL. • Deep RL and RLHF: Deep Q-Networks (DQN); policy gradient methods; Reinforcement Learning from Human Feedback (RLHF) — aligning LLMs; applications in games, robotics, and autonomous systems. • NLP fundamentals: language models (n-gram); text classification; information retrieval (TF-IDF); information extraction (NER). • NLP with transformers: attention mechanism; BERT (bidirectional pre-training) and GPT (autoregressive); fine-tuning and prompt engineering; sentiment analysis, summarization, machine translation. **Source:** TB1: Chapters 17, 22–24 (Russell & Norvig); TB2: Chapter 16

	(Nilsson)
AI Integration	Use ChatGPT to trace Q-learning on a grid world for 5 steps; use Claude to explain RLHF and how it is used to align LLMs; use an AI assistant to compare BERT and GPT and identify the right model for a given NLP task.

UNIT – V	**ROBOTICS, COMPUTER VISION, AND AI ETHICS**	**9 Hrs**

Content	□ Robotics: robot hardware — sensors and actuators; robotic perception; motion planning; SLAM (Simultaneous Localization and Mapping); robotic software architectures — ROS. □ Computer vision: image formation; image features (edges, corners); image classification; object detection — YOLO, R-CNN; image segmentation. Deep learning for vision: CNN architectures (AlexNet, ResNet); Vision Transformers (ViT); generative AI — GANs and diffusion models. □ AI ethics and governance: bias and fairness — documented cases (COMPAS, facial recognition disparities); AI safety and alignment; EU AI Act 2024 — risk tiers; India DPDP Act 2023; UNESCO AI Ethics Recommendation; environmental cost of AI. **Source:** TB1: Chapters 25–27, 28 (Russell & Norvig); TB2: Chapter 18 (Nilsson)
AI Integration	Use ChatGPT to explain SLAM with a robot navigation scenario; use Claude to explain YOLO object detection; use an AI assistant to classify 5 real AI applications under EU AI Act risk tiers; use NotebookLM to generate an AI ethics study guide.

Course Outcomes (COs

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Describe AI foundations, agent architectures, environment types, and the modern AI landscape. (L2)	L2 (Understand)
CO2	Apply uninformed, informed, and adversarial search strategies to solve AI problems. (L3)	L4 (Analyse)
CO3	Represent knowledge using propositional and first-order logic and apply planning algorithms. (L3)	L4 (Analyse)
CO4	Explain reinforcement learning, NLP models, and transformer-based AI applications. (L2)	L4 (Analyse)
CO5	Explain robotics and computer vision principles; evaluate AI ethics and governance frameworks. (L4)	L5 (Evaluate)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define AI history, PEAS framework, agent types, and environment properties.
L3	Apply	**CO2, CO3**	Implement BFS, DFS, A*, Hill Climbing; apply propositional logic.
L4	Analyse	**CO2–CO4**	Compare uninformed vs. informed search; analyse Q-learning convergence.
L5	Evaluate	**CO5**	Evaluate robot sensor fusion; assess AI ethics frameworks for a given deployment.
L6	Create	**CO3, CO5**	Design a planning agent with propositional logic rules and an ethics impact assessment.

CO – PO / PSO Articulation Matrix (NBA)

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO1	L2	**3**	**2**	1	**2**	**2**	1	–	1	–	1	–
CO2	L4	**3**	**3**	**2**	**3**	**3**	–	–	–	1	**2**	–
CO3	L4	**3**	**3**	**3**	**3**	**3**	1	–	–	1	**2**	–
CO4	L4	**3**	**3**	**2**	**3**	**3**	1	–	1	1	**2**	1
CO5	L5	**3**	**3**	**3**	**3**	**3**	**2**	1	**2**	1	**2**	1

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO-PO Correlation Highlights	Strong PO2 (Problem Analysis) and PO4 (Investigation) through search algorithm analysis and RL convergence evaluation. PO3 through knowledge representation and planning agent design. PO6 and PO8 (Ethics) through responsible AI, bias in ML systems, and AI ethics impact assessment in Unit V.

LEARNING RESOURCES

Textbooks

1.TB1: Stuart J. Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 4th Edition,Pearson Education, 2020. (Primary — all five units. Chapter previews at aima.cs.berkeley.edu)

2.TB2: Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998. (Units I–IV — search, logic, planning, and learning foundations.)

Reference Books

1.Ethem Alpaydin, Introduction to Machine Learning, 4th Edition, MIT Press, 2020.
2.an Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press. (Free at deeplearningbook.org)

Online Resources

- NPTEL – Introduction to AI, IIT Madras – nptel.ac.in/courses/106106139
- UC Berkeley CS188 – Intro to AI – inst.eecs.berkeley.edu/~cs188
- AIMA Python Code – github.com/aimacode/aima-python | HuggingFace NLP Course –huggingface.co/learn/nlp-course
- EU AI Act – eur-lex.europa.eu | India DPDP Act 2023 – meity.gov.in | UNESCO AI Ethics – unesco.org

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR (AUTONOMOUS)

II B.Tech - II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS3105) INTELLIGENT DATA PROCESSING AND SYSTEM INTEGRATION LAB

Lab Course Outcomes

- Implement REST API integration with retry-backoff logic and Kafka-based streaming ingestion; build an ETL and ELT pipeline using Python. (L3)
- Build intelligent Pandas processing pipelines with method chaining and validation; implement Spark DataFrame transformations on a large dataset. (L4)
- Integrate PostgreSQL using SQLAlchemy ORM, MongoDB using pymongo, and FAISS vector index from Python; implement a hybrid search pipeline. (L3)
- Design and implement FastAPI REST endpoints and a gRPC service; apply circuit breaker and API gateway patterns for AI microservice integration. (L4)
- Build and schedule Apache Airflow DAGs with branching and XCom; instrument a pipeline with structured logging, Prometheus metrics, and GDPR/DPDP compliance. (L5)

List of Experiments

Unit I	Data Ingestion and System Integration Patterns

1. REST API Integration with Retry-Backoff and Webhook Receiver

Implement a REST API client using the requests library to consume a public API (e.g., OpenWeatherMap or JSONPlaceholder). Requirements: (a) GET with query parameters and response pagination — fetch all pages and aggregate results; (b) POST with JSON body and Bearer token authentication; (c) exponential retry-with-backoff using the tenacity library — configure max_attempts=5, wait_exponential(min=1, max=30), retry on requests.exceptions.ConnectionError and HTTP 429/503; (d) implement a simple webhook receiver using Flask — expose a POST /webhook endpoint, log incoming payloads to a JSON file, and return HTTP 200. Test by sending 10 simulated webhook calls using requests.post(). Use ChatGPT to generate the retry decorator configuration — verify that it correctly retries on status 429 and document the AI output vs. your implementation.

2. Kafka Producer-Consumer Pipeline and ETL vs. ELT Comparison

Set up Apache Kafka locally (using Docker: docker-compose with Zookeeper + Kafka). Part A — Kafka Producer-Consumer: (a) write a Python producer (kafka-python) that publishes 100 JSON messages to a topic orders (each: {order_id, product, amount, timestamp}); (b) write a consumer that reads from orders, deserializes each message, and appends to a Parquet file using PyArrow; (c) demonstrate consumer group behaviour by running 2 consumers in the same group — show message partitioning. Part B — ETL vs. ELT: (a) implement ETL: extract from a CSV, transform (clean nulls, normalise strings, derive a new column), load into SQLite; (b) implement ELT: load the raw CSV into DuckDB as-is, then apply a dbt-style SQL transformation inside DuckDB; (c) measure and compare total time for both approaches on a 100K-row dataset. Use Claude to compare the two pipelines — record the response, evaluate it, and document one correction if needed.

3. File-Based Integration and S3-Compatible Object Storage

Part A — SFTP simulation: use paramiko to connect to a local SFTP server (sftpserver Docker image); upload 5 CSV files, list the remote directory, download a specific file, and handle FileNotFoundError gracefully. Part B — S3-compatible object storage using MinIO (Docker): (a) set up MinIO locally; (b) use boto3 to: create a bucket, upload a 10 MB Parquet file, list bucket contents with metadata, download the file, and delete it; (c) implement multipart upload for a 100 MB file using boto3 multipart API; (d) enable server-side encryption (SSE-S3) for the upload. Part C — Integration pattern: implement the adapter pattern — write a StorageAdapter abstract class with upload(), download(), list() methods; implement SFTPAdapter and S3Adapter that satisfy the same interface; demonstrate that switching from SFTP to S3 requires only changing the adapter without touching the calling code. Use an AI assistant to generate the abstract adapter class — review and implement it.

Unit II	**Intelligent Data Processing**

4. NumPy Vectorised Processing and Performance Benchmarking

Benchmark vectorised vs. loop implementations across 5 operations on arrays of sizes [10K, 100K, 1M, 10M]: (a) element-wise square root; (b) computing row-wise dot product between two 2D arrays; (c) batch normalisation: (X − mean) / std using broadcasting; (d) computing a cosine similarity matrix between 1000 vectors of dimension 128 using np.dot and manual loop — compare times; (e) solving a linear system Ax = b for 100 different right-hand sides using np.linalg.solve vs. loop. Plot time vs. array size for all 5 on a single Matplotlib figure with log scale. Demonstrate memory-efficient dtype selection: compute mean absolute error between float32 and float64 results for the batch normalisation; report memory usage with arr.nbytes. Use ChatGPT to explain why NumPy broadcasting avoids the loop — verify with 1 counter-example where broadcasting raises an error, and document.

5. Pandas Intelligent Pipeline with Method Chaining, Validation, and Audit Log

Load the NYC Taxi Trip dataset (or equivalent with 500K+ rows). Build a fully method-chained Pandas pipeline using .pipe(): Stage 1 — clean: drop exact duplicates, clip trip_distance to [0.1, 100], fill null passenger_count with median; Stage 2 — validate: assert no nulls in required columns, assert all fare_amount> 0, raise DataQualityError if >5% rows fail; Stage 3 — enrich: add hour_of_day, day_of_week, is_weekend, log1p_fare_amount, haversine_distance (from pickup/dropofflat-lon); Stage 4 — aggregate: groupbypickup_hour and passenger_count with multiple aggregations (mean, median, count, std) using pd.NamedAgg. Build a processing audit log: record input row count, output row count, null rate per column before/after, processing time per stage, and write to JSON. Apply merge_asof() for a time-series join: merge the trip dataset with a weather_dataDataFrame (temperature per 30-minute interval) on timestamp. Use Claude to identify 2 performance bottlenecks in the pipeline — implement the suggested optimisations, measure improvement, and document.

6. Apache Spark — Distributed DataFrame Processing and UDFs

Initialise a local SparkSession with 4 worker threads. Load a 1M-row CSV (generated with Faker or downloaded) into a Spark DataFrame. Demonstrate lazy evaluation: build a transformation chain of 5 operations (filter, withColumn, groupBy, join, sort) without calling an action; use df.explain() to print the physical plan; observe the DAG in the Spark UI. Transformations: (a) filter trips where distance > 2; (b) withColumn to add

fare_per_km = fare / distance; (c) groupBy hour and compute mean fare, count, and percentile_approx(fare, 0.95) using Window; (d) join with a zones lookup DataFrame on location_id; (e) write output to Parquet with partitionBy('hour'). Implement a Spark UDF for custom text processing: clean address strings (lowercase, remove punctuation, strip) — compare UDF performance with a built-in Spark SQL function equivalent. Run the same aggregation using Spark SQL syntax (spark.sql(...)). Use ChatGPT to explain the difference between a narrow transformation and a wide transformation — classify each of your 5 transformations and verify against the explain() output.

Unit III	Database Integration for AI Systems

7. SQLAlchemy ORM — Full CRUD, Connection Pooling, and Migrations

Design a data model for an ML experiment tracker: Experiment (id, name, created_at), Run (id, experiment_id FK, status, started_at, ended_at), Metric (id, run_id FK, key, value, step), Artifact (id, run_id FK, path, size_bytes). Implement using SQLAlchemy ORM (declarative base, relationship, backref) targeting PostgreSQL (use Docker for a local instance). Operations: (a) insert 5 experiments each with 3 runs and 10 metrics per run using bulk_insert_mappings for performance; (b) query: find the top-3 runs by max accuracy metric using joined query with subquery; (c) upsert: use on_conflict_do_update (PostgreSQL INSERT ... ON CONFLICT) to update an existing run's status; (d) configure QueuePool (pool_size=5, max_overflow=10, pool_timeout=30) and demonstrate concurrent inserts from 5 threads; (e) implement Alembic migrations: add a tags column to Experiment, generate a migration script with alembic revision --autogenerate, apply with alembic upgrade head, verify. Use Claude to design the repository pattern for this schema — implement a RunRepository class with 4 methods.

8. MongoDB, Redis, and NoSQL Integration for AI Data Patterns

MongoDB (pymongo): (a) design a document schema for AI model metadata: {model_id, name, framework, hyperparameters: {}, metrics: {}, tags: [], created_at}; insert 30 documents; (b) aggregation pipeline: group by framework, compute avg accuracy, max f1; (c) text index on name + tags, demonstrate $text search; (d) upsert 10 documents using update_one with upsert=True; (e) demonstrate change streams: open a change stream watch cursor, insert 3 documents in a separate thread, print the change events. Redis (redis-py): (a) key-value: cache ML model metadata with TTL=3600; implement cache-aside pattern — check cache, miss → query MongoDB, set cache; (b) sorted set: implement a leaderboard of top models by F1 score using ZADD/ZREVRANGE; (c) Pub/Sub: publish 10 training progress events on channel ml:progress, consume in a subscriber thread and print. Use ChatGPT to explain the MongoDB aggregation pipeline stages $group, $project, $match — verify all 3 claims against your experimental output and document discrepancies.

9. Vector Database Integration — FAISS, Chroma, and Hybrid Search

Generate 1000 product descriptions; encode using sentence-transformers (all-MiniLM-L6-v2) to get 384-dim embeddings. FAISS: (a) build IndexFlatL2 and IndexIVFFlat (nlist=20); (b) for 20 queries measure Recall@5 and latency for both; (c) serialise the IVF index to disk (faiss.write_index) and reload (faiss.read_index) — verify results are identical; (d) implement a re-ranking step: FAISS retrieves top-20, then re-rank by BM25 score on product title. Chroma: (a) create a persistent collection, add 1000 embeddings with metadata (category, price, in_stock); (b) vector query — top-5 by similarity; (c) hybrid search — vector similarity AND metadata filter (category=Electronics AND price < 5000 AND in_stock=True); (d) update 10 documents and delete 5 — verify collection

count. Implement the repository pattern: VectorStoreRepository abstract class with add(), search(), delete() methods; FAISSRepository and ChromaRepository implementations — swap with one line change. Use an AI assistant to explain when to use IVF vs. HNSW index — design 1 experiment to test the recommendation.

Unit IV	API Design and Microservice Integration

10. FastAPI — Full REST API with Pydantic, Auth, Middleware, and Background Tasks

Build a FastAPI application exposing a machine learning prediction service. Endpoints: (a) POST /predict: accept {text: str, model_version: str}, validate with Pydantic (text min_length=1, max_length=1000, model_version pattern=r'v\d+'), run a mock classification (random label + confidence), return {label, confidence, latency_ms}; (b) GET /models: return list of available model versions from an in-memory dict; (c) POST /batch_predict: accept List[PredictRequest] (max 100), process in a background task using BackgroundTasks, return {job_id}; (d) GET /jobs/{job_id}: return job status and results when complete. Middleware: add RequestLoggingMiddleware (log method, path, status, latency to a JSON log file) and RateLimitMiddleware (max 10 requests/second per client IP using a sliding window counter in a dict). Security: implement OAuth2 password flow using fastapi.security.OAuth2PasswordBearer — protect /predict with a valid token check. Lifespan: load the "model" (a sklearn LogisticRegression trained on a toy dataset) on startup. Document all endpoints using FastAPI's built-in /docs. Use ChatGPT to design the Pydantic request and response models — review for missing validation rules and implement corrections.

11. gRPC Service and Circuit Breaker Pattern

Part A — gRPC: define a protobuf schema (prediction.proto) with: PredictRequest {text: string, model_id: string}, PredictResponse {label: string, confidence: float, latency_ms: int64}, and a service PredictionService with 3 RPCs: Predict (unary), PredictStream (server-streaming — stream progress updates during batch prediction), PredictBatch (client-streaming — send multiple texts, receive one aggregated response). Generate Python stubs using grpc_tools.protoc. Implement the server and a client. Benchmark: compare gRPC unary vs. REST (FastAPI) for 1000 identical prediction requests — measure latency (p50, p95, p99) and throughput (req/s). Part B — Circuit Breaker: implement a circuit breaker around a downstream service call (simulate failures by randomly raising exceptions 30% of the time): (a) use the circuitbreaker library with FAILURE_THRESHOLD=3, RECOVERY_TIMEOUT=10; (b) demonstrate the 3 states — Closed (normal), Open (fail-fast), Half-Open (probe) — by forcing failures and observing state transitions; (c) implement graceful degradation: when circuit is Open, return a cached result instead of failing. Use Claude to explain the circuit breaker state machine — map each state to 1 of your experimental observations.

12. Async FastAPI, API Gateway Pattern, and Inter-Service Communication

Build 3 microservices: ServiceA (data-ingestion — accepts raw text), ServiceB (prediction — returns label + confidence), ServiceC (logging — stores request+response to SQLite). Each is a separate FastAPI app running on different ports. Async: implement async endpoints using async def and await; use aiohttp.ClientSession for HTTP calls between services (not requests); demonstrate that concurrent requests complete faster — send 50 concurrent requests using asyncio.gather() and compare with synchronous calls. API Gateway (mock): implement a 4th FastAPI service that acts as a gateway — receives a client request, authenticates the token, routes to ServiceA or ServiceB based on the path,

applies rate limiting (10 req/s per IP), and logs to ServiceC via async call. Message-passing integration: ServiceB publishes prediction results to a Kafka topic predictions; ServiceC subscribes and persists. Demonstrate end-to-end flow: client → Gateway → ServiceA → ServiceB (Kafka publish) → ServiceC (Kafka consume → SQLite). Use an AI assistant to generate the aiohttp client session code for ServiceA-to-ServiceB communication — test and document corrections.

Unit V	Workflow Orchestration and Responsible Integration

13. Apache Airflow — DAG Design, Branching, XCom, and Data Quality Gates

Install Airflow locally (pip install apache-airflow) and initialise the metadata database. Build a DAG ml_pipeline_dag (schedule='0 2 * * *', catchup=False) with: Task 1: PythonOperator — ingest_data: download a CSV from a URL, save to /tmp/raw.csv, push filename to XCom; Task 2: PythonOperator — validate_data: pull filename from XCom (ti.xcom_pull), check null rates, row count, schema; push validation_status (pass/fail); Task 3: BranchPythonOperator — branch_on_quality: read validation_status from XCom; if pass, go to process_data; if fail, go to alert_team; Task 4a: process_data — clean and transform, save to /tmp/processed.parquet; Task 4b: alert_team — BashOperator that echoes an alert message and exits; Task 5: ShortCircuitOperator — check_model_threshold: only continue if accuracy > 0.80 (read from XCom); Task 6: PythonOperator — register_model: simulate MLflow model registration. Configure retries=2, retry_delay=timedelta(minutes=1) on ingest_data and process_data. Embed a Great Expectations checkpoint inside validate_data: validate that null rate < 5% for all columns. Use ChatGPT to explain XCom serialisation limits — demonstrate the limit by attempting to push a large DataFrame and document the error and solution.

14. Pipeline Observability — Structured Logging, Prometheus Metrics, and OpenTelemetry Tracing

Instrument the FastAPI prediction service from Experiment 10 with full observability. Structured logging: configure Python logging with a JSON formatter (python-json-logger); log every request with: request_id (UUID), endpoint, method, status_code, latency_ms, model_version, client_ip; write to logs/app.jsonl; demonstrate log parsing using pandas.read_json(lines=True) and filtering for errors (status >= 400). Prometheus metrics: add prometheus-client to FastAPI; expose GET /metrics; define: (a) prediction_requests_total (Counter, labels: endpoint, status, model_version); (b) prediction_latency_seconds (Histogram, buckets=[0.01, 0.05, 0.1, 0.25, 0.5, 1.0]); (c) active_connections (Gauge). Send 200 requests with varied latency (random.uniform(0.01, 0.5)) and verify metrics at /metrics. OpenTelemetry: instrument with opentelemetry-sdk and opentelemetry-instrumentation-fastapi; create a trace for each request with 2 child spans: preprocess_input and run_model; export traces to OTLP (Jaeger or console). Demonstrate distributed tracing: trace a request from the API Gateway (Exp 12) through to ServiceB, showing the trace propagated via HTTP headers. Use Claude to generate the JSON logging configuration — implement it and verify that all required fields appear in every log line.

15. Responsible Integration — PII Detection, Consent-Aware Routing, and GDPR/DPDP Compliance

PII detection pipeline: (a) build a PII detector using spaCy NER (en_core_web_sm) combined with regex patterns for email (RFC 5322), Indian mobile number (10 digits starting with 6–9), Aadhaar (12 digits with space every 4), and PAN (ABCDE1234F pattern); (b) load a synthetic dataset of 500 customer records containing PII; detect and log

PII fields per record; (c) implement a PII redaction function: replace detected PII with [REDACTED-<TYPE>] before persistence; verify that redacted records contain no raw PII. Consent-aware data routing: (a) augment records with consent flags: {analytics_consent: bool, marketing_consent: bool, third_party_consent: bool}; (b) implement a DataRouter class with route(record) method: if analytics_consent=True, route to analytics_db (SQLite); if marketing_consent=True, route to marketing_db; if neither, route to minimal_storage_db (store only id, timestamp); (c) process all 500 records and verify each reached the correct destination. Right-to-erasure: implement delete_user(user_id) that: removes all records from all 3 SQLite databases where user_id matches; logs the deletion with timestamp and operator; verify with a post-deletion query. GDPR/DPDP checklist: produce a 15-item compliance checklist (purpose limitation, data minimisation, storage limitation, accuracy, security, right-to-access, right-to-erasure, consent withdrawal, cross-border transfer, DPO appointment, breach notification, DPDP Act 2023 data fiduciary obligations, children's data, automated decision-making, audit trail) — assess each as Implemented / Partial / Not Implemented for this experiment. Use an AI assistant to generate the DPDP Act 2023 obligations checklist — evaluate completeness against the official Act text (meity.gov.in) and document gaps.

AI Tools Integration (Lab)	ChatGPT / Claude for code generation, architecture design, pattern explanation, and checklist creation (all AI outputs independently verified and documented). kafka-python for Kafka producer-consumer. SQLAlchemy + Alembic for ORM and migrations. pymongo + redis-py for NoSQL integration. FAISS and Chroma for vector similarity search. sentence-transformers (HuggingFace) for embeddings. FastAPI + gRPC for API development. aiohttp for async HTTP. Apache Airflow (pip) for DAG orchestration. Great Expectations for inline data validation. Prometheus-client + OpenTelemetry for observability. spaCy (en_core_web_sm) for PII detection. MinIO + boto3 for S3-compatible storage. circuitbreaker for circuit breaker pattern.

Course Outcomes (COs

After successful completion of the lab, the student will be able to:

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Implement REST API integration with retry-backoff logic and Kafka-based streaming ingestion; build an ETL and ELT pipeline using Python. (L3)	L3 (Apply)
CO2	Build intelligent Pandas processing pipelines with method chaining and validation; implement Spark DataFrame transformations on a large dataset. (L4)	L4 (Analyse)
CO3	Integrate PostgreSQL using SQLAlchemy ORM, MongoDB using pymongo, and FAISS vector index from Python; implement a hybrid search pipeline. (L3)	L3 (Apply)
CO4	Design and implement FastAPI REST endpoints and a gRPC service; apply circuit breaker and API gateway patterns for AI microservice integration. (L4)	L4 (Analyse)
CO5	Build and schedule Apache Airflow DAGs with branching and XCom; instrument a pipeline with structured logging, Prometheus metrics, and GDPR/DPDP compliance. (L5)	L5 (Evaluate)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L3	Apply	**CO1–CO3**	Implement Kafka producer-consumer pipeline; build Pandas method-chaining pipeline; integrate FAISS vector index from Python; build SQLAlchemy ORM CRUD operations.
L4	Analyse	**CO2, CO4**	Analyse Spark lazy evaluation DAG; benchmark gRPC vs. REST latency; compare FAISS Flat vs. IVF recall; measure circuit breaker state transitions.
L5	Evaluate	**CO5**	Evaluate Airflow DAG branching logic; assess GDPR/DPDP Act 2023 compliance checklist; evaluate Prometheus metrics for SLO adherence.
L6	Create	**CO4, CO5**	Design a complete async multi-service pipeline (Gateway + ServiceA + ServiceB + Kafka + ServiceC); implement PII detection + consent-aware routing + right-to-erasure system.

CO – PO / PSO Articulation Matrix (NBA)

Scale: 3 – Strong | 2 – Moderate | 1 – Slight | – No Correlation

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgm
CO1	L3	**3**	**2**	**2**	**2**	**3**	–	–	–	–	1	–
CO2	L4	**3**	**3**	**2**	**2**	**3**	–	–	–	1	1	–
CO3	L3	**3**	**3**	**2**	**2**	**3**	–	–	–	1	1	–
CO4	L4	**3**	**3**	**3**	**2**	**3**	1	–	–	1	**2**	1
CO5	L5	**3**	**3**	**3**	**3**	**3**	**2**	1	**2**	1	**2**	1

CO-PO Correlation Highlights	Strong PO5 (Modern Tools) through kafka-python, SQLAlchemy, pymongo, FAISS, Chroma, FastAPI, gRPC, aiohttp, Airflow, Prometheus, OpenTelemetry, spaCy, and diffprivlib across all 15 experiments. PO3 (Design/Development) through API gateway design, repository pattern implementation, and Airflow DAG orchestration. PO6 (Engineer & Society) and PO8 (Ethics) strongly addressed in Experiment 15 through PII detection, consent-aware data routing, GDPR Article 5 and India DPDP Act 2023 right-to-erasure implementation.

LEARNING RESOURCES

Textbooks

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017.
2. Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022.
3. Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022.
4. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson.

Reference Books

1. Sam Newman, Building Microservices, 2nd Edition, O'Reilly Media, 2021.
2. Bas Geerdink, A Practical Guide to Building Ethical AI Systems, Manning, 2023.

Online Resources

- Apache Airflow – airflow.apache.org |FastAPI – fastapi.tiangolo.com |gRPC Python – grpc.io/docs/languages/python
- SQLAlchemy – sqlalchemy.org | Apache Spark – spark.apache.org |kafka-python – kafka-python.readthedocs.io
- FAISS – github.com/facebookresearch/faiss | Chroma – trychroma.com |HuggingFace – huggingface.co
- Prometheus Python – prometheus.io/docs |OpenTelemetry Python – opentelemetry.io/docs/languages/python
- India DPDP Act 2023 – meity.gov.in | GDPR – gdpr.eu |spaCy – spacy.io

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0516) INTELLIGENT DATABASE MANAGEMENT SYSTEMS LAB
(Professional Core – V)
(Common to CSE & Allied branches)

• Prerequisites: **Data Structures**

Course Objectives:

The objectives of this course is to

1. *Develop practical skills in database design and implementation through ER modelling, relational schema design, and database creation for real-world applications.*
2. *Apply SQL techniques for database operations including data definition, manipulation, querying, optimization, indexing, and transaction control.*
3. *Design efficient and reliable database systems using normalization, schema decomposition, concurrency control, and recovery mechanisms.*
4. *Explore modern intelligent database technologies including NoSQL databases, graph databases, vector databases, and AI-enabled data management systems.*
5. *Utilizer Artificial Intelligence tools and platforms to support database design, natural language querying, schema validation, query optimization, and intelligent analytics.*

List of Experiments

MODULE I: Database Modelling And Sql Foundations

AI Integration: Use draw.io and ChatGPT to generate and validate ER diagrams and schema designs.

Suggested Tools: draw.io; ChatGPT

Experiments:

1. **Experiment 1: ER Modelling —** Design an ER diagram for a real-world application (e.g., library, hospital, or e-commerce system) and convert it into a relational schema using ER-to-relational mapping rules.
2. **Experiment 2: Database & DDL Operations —** Create a database and perform DDL operations (CREATE, ALTER, DROP) with integrity constraints such as Primary Key, Foreign Key, CHECK, UNIQUE, and NOT NULL.
3. **Experiment 3: DML & DQL Operations —** Perform DML and DQL commands (INSERT, UPDATE, DELETE, SELECT) with filtering, sorting, grouping, and aggregate functions on the created schema.

MODULE II: Sql Programming And Query Optimization

AI Integration: Use GitHub Copilot and ChatGPT to generate SQL queries and compare execution plans.

Suggested Tools: GitHub Copilot; ChatGPT

Experiments:

1. **Experiment 4: Joins & Subqueries —** Develop SQL queries using joins, nested queries, subqueries, set operations, aggregate functions, GROUP BY, and HAVING clauses.
2. **Experiment 5: Views, Indexes & Optimization —** Create and analyze views and indexes, and perform query optimization comparing indexed and non-indexed access methods.
3. **Experiment 6: Relational Algebra to SQL —** Implement relational algebra operations and convert relational algebra expressions into equivalent SQL statements.

MODULE III : Database Design And Normalization

AI Integration: Use ChatGPT and Jupyter Notebook to assist normalization and validate using Armstrong's axioms.

Suggested Tools: ChatGPT; Jupyter Notebook

Experiments:

1. **Experiment 7: Normalization (1NF–BCNF) —** Identify functional dependencies and perform normalization of relations from 1NF through BCNF.
2. **Experiment 8: Closures & Decomposition —** Implement attribute closure computation, minimal cover generation, lossless decomposition, and dependency preservation.
3. **Experiment 9: Schema Comparison —** Compare normalized and denormalized schema designs for analytical workloads and document the trade-offs.

MODULE IV : Transaction Management And Recovery

AI Integration: Use Python simulations and ChatGPT to generate transaction schedules and precedence graph analysis.

Suggested Tools: Python; ChatGPT

Experiments:

1. **Experiment 10: Transaction Control —** Implement transaction control operations using COMMIT, ROLLBACK, and SAVEPOINT.
2. **Experiment 11: Concurrency Control —** Simulate concurrency control mechanisms including locking protocols, two-phase locking, deadlock detection, and serializability analysis.
3. **Experiment 12: Logging & Recovery —** Implement database recovery mechanisms including logging, checkpointing, undo-redo recovery, and Write Ahead Logging.

MODULE V: Intelligent Data Management For Ai Applications

AI Integration: Use ChatGPT, LangChain, MongoDB Atlas, FAISS, Chroma, and Neo4j for intelligent database experimentation.

Suggested Tools: ChatGPT; LangChain; MongoDB Atlas; FAISS; Chroma; Neo4j

Experiments:

1. **Experiment 13: NoSQL with MongoDB —** Perform CRUD operations and aggregation queries using a NoSQL database such as MongoDB.
2. **Experiment 14: Graph Databases with Neo4j —** Develop a graph database application using Neo4j and implement Cypher queries.
3. **Experiment 15: Intelligent Applications —** Build intelligent database applications using Text-to-SQL, RAG-over-SQL, vector databases (FAISS/Chroma), similarity search, and AI-assisted data analytics.

Suggested Mini Projects:

1. An AI-assisted Text-to-SQL query interface for a chosen database schema.
2. A hybrid application combining a relational database (SQL) with a vector database (FAISS/Chroma) for semantic search.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Design ER models and convert them into relational schemas for real-world applications.	L3 (Apply)
CO2	Construct and execute SQL queries for data definition, manipulation, retrieval, and optimization.	L3 (Apply)
CO3	Apply normalization techniques and schema decomposition methods to design efficient databases.	L3 (Apply)
CO4	Implement transaction management, concurrency control, and recovery mechanisms in database environments.	L4 (Analyze)
CO5	Develop intelligent database applications using NoSQL systems, vector databases, and AI-assisted querying techniques.	L4 (Analyze)

CO – PO Articulation Matrix :

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	Bloom's Level	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	L3	3	2	3	2	3	–	–	–	1	1	–	2
CO2	L3	3	2	3	2	3	–	–	–	1	1	–	2
CO3	L3	3	3	2	2	3	–	–	–	1	1	–	2
CO4	L4	3	3	3	2	3	–	–	–	1	1	–	2
CO5	L4	3	3	3	3	3	–	–	1	2	2	1	3

Textbooks :

1. Silberschatz, A., Korth, H. F., & Sudarshan, S., *Database System Concepts*, 7th Ed., McGraw-Hill Education, 2020.
2. Elmasri, R., & Navathe, S. B., *Fundamentals of Database Systems*, 7th Ed., Pearson, 2016.

Reference Books :

1. Ramakrishnan, R., & Gehrke, J., *Database Management Systems*, 3rd Ed., McGraw-Hill Education, 2003.
2. Kleppmann, M., *Designing Data-Intensive Applications*, O'Reilly Media, 2017.

Online resources :

1. NPTEL – Database Management System – nptel.ac.in
2. MongoDB University – Free NoSQL Courses – university.mongodb.com
3. Neo4j GraphAcademy – Free Graph Database Courses – graphacademy.neo4j.com

Recommended Free & Open-Source AI/Computational Tools

1. **AI Assistants:** ChatGPT; GitHub Copilot – query generation, debugging, and schema/normalization validation.
2. **Modelling Tools:** draw.io – ER diagram design.
3. **Databases & Engines:** MongoDB (Atlas free tier); Neo4j (CommUnity/AuraDB Free) – NoSQL and graph database experimentation.
4. **Vector Search Libraries:** FAISS; Chroma – similarity search and vector database applications.
5. **Frameworks:** LangChain – building Text-to-SQL and RAG-over-SQL pipelines.
6. **Notebooks & Simulation:** Python & Jupyter Notebook – transaction simulations and normalization/closure computations.
7. TensorFlow / Keras (free, open-source) – For optional AI demonstrations in regression and classification tasks.

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

L	T	Pr	P	C
1	0	0	2	2

II B.Tech – II Sem

(26CS4771) DATA ANALYTICS USING PYTHON
(Skill Enhancement Course – II)
(Common to CSE & Allied branches)

Course Objectives :

The objectives of this course is to

1. *Develop practical skills for handling, processing, and analyzing data using Python-based analytical tools and libraries.*
2. *Enable students to perform data cleaning, transformation, aggregation, and exploratory analysis on structured datasets.*
3. *Build competency in creating visualizations and dashboards for effective interpretation and communication of analytical insights.*
4. *Introduce real-world analytics workflows involving data acquisition, processing, reporting, and presentation.*
5. *Prepare students with a strong foundation in data analytics for advanced courses in Artificial Intelligence, Machine Learning, and Data Science.*

List of Experiments

MODULE I: Data Handling Using Python

Topics: Data Analytics Workflow; Jupyter Notebook Environment; Data Sources and File Formats; Importing Data; CSV, Excel and JSON Processing; NumPy Arrays; Data Manipulation; Vectorized Operations; Data Export.

AI Integration: Dataset understanding, automatic profiling and analytical assistance.

Suggested Tools: Python, Jupyter Notebook, Pandas, NumPy

Experiments:

1. **Experiment 1: Dataset Import and Exploration —** Import datasets from CSV, Excel and JSON formats. Examine dimensions, data types, missing values and generate summary statistics.
2. **Experiment 2: Numerical Data Processing using NumPy —** Perform array creation, indexing, slicing, aggregation, mathematical operations and vectorized computations.
3. **Experiment 3: DataFrame Operations using Pandas —** Create DataFrames and perform selection, filtering, indexing, sorting and aggregation.

MODULE II: Data Processing and Transformation Using Pandas

Topics: Series and DataFrames; Data Selection; Indexing and Filtering; Missing Data Handling; Data Cleaning; Aggregation; Grouping; Merging; Sorting; Feature Construction (without ML).

AI Integration: AI-assisted cleaning, transformation recommendations and optimization.

Suggested Tools: Pandas, Jupyter Notebook

Experiments:

1. **Experiment 4: Data Cleaning and Missing Value Handling —** Identify duplicate records, missing values and outliers. Apply cleaning and preprocessing techniques.
2. **Experiment 5: Data Transformation and Feature Construction —** Perform grouping, merging, aggregation, reshaping and derived column generation.
3. **Experiment 6: Descriptive Statistics and Data Analysis —** Generate statistical summaries and perform exploratory analysis including Mean, Median, Mode, Variance and Distribution Analysis.

MODULE III: Exploratory Data Analysis And Visualization

Topics: Descriptive Statistics; Data Exploration; Correlation Analysis; Data Visualization Principles; Line Charts; Bar Charts; Scatter Plots; Histograms; Boxplots; Heatmaps; Interactive Visualization.

AI Integration: Visualization recommendation, dashboard generation and insight generation.

Suggested Tools: Matplotlib, Seaborn, Plotly

Experiments:

1. **Experiment 7: Data Visualization using Matplotlib —** Generate visual reports using line charts, bar charts, histograms and pie charts.
2. **Experiment 8: Advanced Visualization using Seaborn —** Develop scatter plots, boxplots, heatmaps and distribution analysis.
3. **Experiment 9: Interactive Visualization and Dashboard Design —** Build interactive visualizations and analytical dashboards.
4. **Experiment 10: Correlation and Comparative Data Analysis —** Perform comparative analysis and identify relationships among variables.

MODULE IV: Data Analytics Using Real-World Datasets

Topics: Data Collection Methods; Public Dataset Usage; JSON Processing; API-based Data Collection; Time-Series Data Handling; Data Transformation; Report Generation; Dashboard Concepts.

AI Integration: Automated analytics, report generation and insight extraction.

Suggested Tools: Pandas, Requests, Plotly

Experiments:

1. **Experiment 11: Data Collection using APIs and JSON Processing —** Collect data using APIs and process JSON data.
2. **Experiment 12: Time-Series and Trend Analysis —** Analyze temporal datasets and generate trend reports including date handling, rolling statistics and visualization.
3. **Experiment 13: Analytical Report Generation —** Generate analytical summaries and reports from datasets.

MODULE V: Applied Data Analytics Project

Topics: Analytics Workflow; Data Preparation; Data Visualization; Insight Generation; Dashboard Development; Documentation; Project Presentation.

AI Integration: Data profiling, visualization recommendation, reporting and documentation.

Suggested Tools: Python, Jupyter Notebook, Plotly

Experiments:

1. **Experiment 14: Dashboard Development using Real-world Datasets —** Create complete dashboards for decision support using Education, Healthcare, Retail or Weather datasets.
2. **Experiment 15: Applied Data Analytics Mini Project —** Develop and present a complete analytics solution including data collection, data cleaning, analysis, visualization, dashboard and report generation.

Suggested Mini Projects:

1. Student Performance Analytics
2. Retail Sales Analytics
3. Healthcare Analytics
4. Weather Analytics
5. Financial Data Analytics
6. Social Media Analytics
7. Sports Analytics
8. Survey Data Analytics

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Import, process, and manipulate datasets using Python analytical libraries.	L4 (Analyze)
CO2	Apply data cleaning, transformation, and aggregation techniques to prepare datasets for analysis.	L5 (Evaluate)
CO3	Perform exploratory data analysis and create meaningful visualizations to derive insights.	L5 (Evaluate)
CO4	Develop analytical reports and dashboards using Python tools and AI-assisted analytics workflows.	L6 (Create)
CO5	Design and implement data-driven analytical solutions using real-world datasets.	L6 (Create)

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight, – No Correlation

CO	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning
CO1	3	2	2	2	3	–	–	–	1	1	–	2
CO2	3	3	2	2	3	–	–	–	1	1	–	2
CO3	3	3	2	3	3	–	–	–	1	2	1	2
CO4	3	3	3	3	3	–	–	–	1	3	2	3
CO5	3	3	3	3	3	1	–	1	2	3	3	3

Textbooks :

1. Wes McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy and Jupyter*, O'Reilly Media.
2. Jake VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media.
3. Joel Grus, *Data Science from Scratch: First Principles with Python*, O'Reilly Media.

Reference books :

1. Cathy O'Neil and Rachel Schutt, *Doing Data Science*, O'Reilly Media.
2. Trevor Hastie, Robert Tibshirani and Gareth James, *An Introduction to Statistical Learning*, Springer.
3. Allen B. Downey, *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media.
4. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, O'Reilly Media (selected chapters on data preparation and analysis).
5. Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press (reference for analytics foundation).

Online learning resources :

1. Python Documentation – docs.python.org
2. Jupyter Documentation – docs.jupyter.org
3. NumPy Documentation – numpy.org/doc
4. Pandas Documentation – pandas.pydata.org/docs
5. Matplotlib Documentation – matplotlib.org/stable
6. Seaborn Documentation – seaborn.pydata.org
7. Plotly Documentation – plotly.com/python

Recommended Free & Open-Source AI/Computational Tools

1. **Core Libraries:** Python, Pandas, NumPy – data handling, manipulation and numerical computing.
2. **Visualization:** Matplotlib, Seaborn, Plotly – static and interactive charts and dashboards.
3. **Environment:** Jupyter Notebook – interactive analysis, experimentation and reporting.
4. **Data Access:** Requests – API-based data collection and JSON processing.
5. **AI Assistants:** ChatGPT, Claude, GitHub Copilot – code generation, automatic data profiling and analytical assistance.

R 26

B.Tech - CAI

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

II B.Tech , II Sem.

L	T	Pr	P	C
2	0	0	0	0

(26HS0812) ENVIRONMENTAL SCIENCE
(Common to all branches)

Prerequisites: No specific prerequisites; open to all engineering branches.

Course Objectives

The objectives of this course are

1. Create awareness about environmental issues, natural resource management and the need for sustainable development.
2. Explain the structure and function of various ecosystems and the significance of biodiversity and its conservation.
3. Describe causes, effects and control of various types of pollution and principles of solid waste management.
4. Examine social, ethical and legislative issues related to environment and sustainable development in India.
5. Understand human population dynamics, environmental health linkages and the role of IT and AI tools in environmental monitoring.

Unit – I	**Multidisciplinary Nature of Environmental Studies and Natural Resources**

Core Content	**Multidisciplinary Nature of Environmental Studies: –** Definition, Scope and Importance – Need for Public Awareness. **Natural Resources:** Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over-exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems – Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies – Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies – Energy resources: Use and exploitation of conventional and non-conventional energy sources, case studies.
AI Integration	AI in Natural Resource Monitoring: Remote sensing and satellite imagery analysis using Google Earth Engine (free cloud platform) for real-time deforestation detection and land use change mapping. Machine learning for groundwater level prediction from historical well data (Python/scikit-learn).

	AI-based precision agriculture – reducing fertilizer and pesticide use through crop health monitoring using drone imagery and computer vision (overview). ML forecasting of renewable energy output (solar irradiance, wind speed) for energy resource planning.
Free / Open-Source AI Tools	Google Earth Engine (free cloud GIS); QGIS (free, open-source); Python (scikit-learn, Matplotlib) – Google Colab (free); Global Forest Watch (free dashboard).

Unit – II	**Ecosystems and Biodiversity and its Conservation**

Core Content	Ecosystems: Concept of an ecosystem – Structure and function of an ecosystem – Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of: Forest ecosystem; Grassland ecosystem; Desert ecosystem; Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries). Biodiversity and its Conservation: Introduction – Definition: genetic, species and ecosystem diversity – Bio-geographical classification of India – Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values – Biodiversity at global, national and local levels – India as a mega-diversity nation – Hot-spots of biodiversity – Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts – Endangered and endemic species of India – Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.
AI Integration	AI in Ecosystem and Biodiversity Studies: AI-powered species identification – applications such as iNaturalist (free) use image recognition to identify plants, animals and insects from photographs, enabling citizen science biodiversity surveys. ML for biodiversity hotspot mapping using satellite imagery and species distribution models (MaxEnt – free). Computer vision for automated wildlife camera trap image analysis – identifying and counting species without manual review. Google Earth Engine for ecosystem health monitoring: NDVI time-series analysis for vegetation cover change detection. AI-assisted assessment of in-situ vs. ex-situ conservation effectiveness.
Free / Open-Source AI Tools	iNaturalist (free app/web); MaxEnt (free species distribution model); Google Earth Engine (free); QGIS (free); Python (rasterio, geopandas) – Google Colab.

Unit – III	Environmental Pollution and Solid Waste Management

Core Content	Environmental Pollution: Definition, cause, effects and control measures of: Air Pollution; Water Pollution; Soil Pollution; Marine Pollution; Noise Pollution; Thermal Pollution; Nuclear Hazards. Solid Waste Management: Causes, effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster Management: floods, earthquake,

	cyclone and landslides.
AI Integration	AI in Pollution Monitoring and Disaster Management: ML-based Air Quality Index (AQI) prediction from meteorological data and traffic density (Python/scikit-learn/TensorFlow – free). IoT sensor networks with AI-driven anomaly detection for real-time water quality monitoring. Computer vision for automated solid waste sorting and classification (overview of AI-enabled sorting robots). OpenAQ platform (free, open data) for accessing global air quality sensor data for ML model training. AI and GIS (QGIS) for disaster risk zone mapping – flood inundation modelling, earthquake vulnerability assessment and cyclone track prediction.
Free / Open-Source AI Tools	OpenAQ (free global AQ data API); QGIS (free); Python (scikit-learn, TensorFlow, Matplotlib) – Google Colab (free); CPCB data portal (free).

Unit – IV	Social Issues and the Environment

Core Content	From unsustainable to sustainable development – Urban problems related to energy – Water conservation, rain water harvesting, watershed management – Resettlement and rehabilitation of people; its problems and concerns, case studies – Environmental ethics: issues and possible solutions – Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust, case studies – Wasteland reclamation – Consumerism and waste products – Environment Protection Act – Air (Prevention and Control of Pollution) Act – Water (Prevention and Control of Pollution) Act – Wildlife Protection Act – Forest Conservation Act – Issues involved in enforcement of environmental legislation – Public awareness.
AI Integration	AI for Climate Change and Sustainable Development: Global climate model data analysis using Python (xarray, Matplotlib) from freely available CMIP6 datasets – visualising temperature rise, precipitation change and sea level scenarios. ML for carbon footprint estimation from energy consumption and transport data (Python/scikit-learn). AI-based watershed management: QGIS + Python for delineation, runoff modelling and rainwater harvesting potential mapping. Overview of AI in environmental compliance monitoring – automated identification of violations using satellite data. Digital tools for public environmental awareness: social media sentiment analysis for pollution perception studies (Python/NLP).
Free / Open-Source AI Tools	Python (xarray, Matplotlib, scikit-learn) – Google Colab (free); CMIP6 climate data (free, ESGF portal); QGIS (free); openLCA (free, LCA software).

Unit – V	Human Population and the Environment

Content	Human Population and the Environment: Population growth, variation among nations – Population explosion – Family Welfare Programmes – Environment and human health – Human Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of Information Technology in Environment and human health – Case studies. Field Work: Visit to a local area to document environmental assets: river / forest / grassland / hill / mountain – Visit to a local polluted site: Urban / Rural / Industrial / Agricultural – Study of common plants, insects and birds – river, hill slopes, etc.
AI Integration	AI in Population Studies and Public Health: ML-based population growth forecasting from census data using Python (time-series regression with scikit-learn / LSTM with TensorFlow). AI in public health surveillance – disease outbreak prediction and mapping using spatiotemporal ML models (overview of WHO FluNet, IDSP data). Role of Information Technology and AI in environmental health: wearable sensors for personal air quality monitoring; smartphone-based water quality testing with ML image analysis. Citizen science apps for field work data collection: iNaturalist, eBird (free) for recording plants, birds and insects during field visits; OpenStreetMap for documenting environmental assets. GIS-based environmental health mapping using QGIS (free).
Free / Open-Source AI Tools	iNaturalist (free); eBird (free); QGIS (free); Python (scikit-learn, TensorFlow) – Google Colab (free); OpenStreetMap (free).

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and explain the multidisciplinary nature of environmental studies, define key concepts of scope and public awareness, and describe the uses, over-exploitation and associated environmental problems of forest, water, mineral, food and energy resources with reference to Indian and global case studies.	L1–L2 (Remember, Understand)
CO2	Describe the structure and function of different ecosystem types, explain ecological succession, food chains and ecological pyramids, and outline the concept and classification of biodiversity, its value, threats and conservation strategies (in-situ and ex-situ) including India's status as a mega-diversity nation.	L2–L3 (Understand, Apply)
CO3	Identify causes, effects and control measures of air, water, soil, marine, noise, thermal and nuclear pollution; apply principles of solid waste management for urban and industrial wastes; analyze pollution case studies and evaluate disaster management strategies for floods, earthquakes, cyclones and	L3–L4 (Apply, Analyze)

	landslides.	
CO4	Analyze social issues related to unsustainable development; evaluate the impacts of climate change, global warming, acid rain and ozone layer depletion; examine Indian environmental legislation (EPA, Air Act, Water Act, Wildlife Protection Act, Forest Conservation Act) and assess their enforcement challenges.	L4–L5 (Analyze, Evaluate)
CO5	Describe population growth trends and their environmental impacts; identify links between environment and human health; demonstrate the role of information technology and AI-based tools in environmental monitoring, public health surveillance and field data collection; apply digital tools (QGIS, Python, iNaturalist) for environmental documentation.	L2–L3 (Understand, Apply)

CO – PO Articulation Matrix

Scale: 3 – Strong, 2 – Moderate, 1 – Slight- No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	–	–	1	3	3	2	–	2	–	2	2	1
CO2	2	2	–	1	2	3	3	2	1	2	–	2	2	2
CO3	2	2	2	2	2	3	3	3	1	2	1	2	2	2
CO4	2	2	2	2	2	3	3	3	1	2	1	3	2	2
CO5	1	1	–	1	3	3	3	2	1	2	–	3	1	2

Text Books:

1. Bharucha E., Textbook of Environmental Studies for Undergraduate Courses, Universities Press.
2. Palaniswamy, Environmental Studies, Pearson Education.

Reference Books

1. Dave D. and Reddy E.S.B., Textbook of Environmental Science, Cengage Publications.
2. Masters G.M. and Ela W.P., Introduction to Environmental Engineering and Science, Prentice Hall of India.

Web Resources & Online Platforms:

- NPTEL: Environmental Science – https://nptel.ac.in/courses/124107002
- Google Earth Engine (free cloud geospatial platform) https://earthengine.google.com
- OpenAQ (free global air quality open data) – https://openaq.org
- iNaturalist (free citizen science biodiversity app) – https://www.inaturalist.org
- QGIS (free, open-source GIS) – https://qgis.org
- India Meteorological Department (free data) – https://mausam.imd.gov.in
- Central Pollution Control Board (CPCB) data portal (free) – https://cpcb.nic.in

Recommended Free & Open-Source AI / Digital Tools:

1. Google Earth Engine (free cloud platform) – real-time satellite data analysis for deforestation, land use and vegetation change detection.
2. QGIS (free, open-source GIS) – environmental mapping, watershed delineation, disaster risk zone mapping, biodiversity hotspot analysis.
3. Python (NumPy, Pandas, Matplotlib, scikit-learn, TensorFlow) on Google Colab (free) – AQI prediction, population growth ML models, climate data analysis.
4. iNaturalist (free, open-source citizen science) – species identification using image recognition during field work; community biodiversity database.
5. eBird (free, Cornell Lab) – bird species recording and mapping during field visits; global bird distribution data.
6. OpenAQ (free, open-source) – global real-time and historical air quality data for ML model training.
7. MaxEnt (free) – species distribution modelling for biodiversity hotspot prediction from occurrence data.
8. Global Forest Watch (free dashboard) – monitoring deforestation and forest cover change using satellite imagery.
9. CMIP6 climate datasets (free, ESGF portal) – global climate model data for Python based climate change scenario analysis.