SRI SAI
INSTITUTE OF TECHNOLOGY AND SCIENCE
AUTONOMOUS

ESTD.:2001

25 YEARS of Academic EXCELLENCE

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Rayachoty – 516 270, Annamayya (Dist.) - (A.P.)

Course Structure and Detailed Syllabi for
I, II, III & IV Year I & II Semesters

B.Tech.
Artificial Intelligence and Data Science

SSITS
R26
Regulations

A Four - Year Regular Degree Program

(Applicable for the batches amitted from the Academic Year 2026 - 27) &

A Three-Year Regular Degree Program under the Lateral Entry Scheme (LES)
(Applicable for the batches admitted from the Academic Year 2027 - 28) academic year 2026-27nwards)

SRI SAI INSTITUTE OF TECHNOLOGY AND SCIENCE
(AUTONOMOUS)
(Approved by AICTE New Delhi & Affiliated to J N T University, Anantapuramu & Accredited by NAAC)
Estd. 2001
Masapeta, RAYACHOTY - 516 270, Annamayya District, A.P.

B. Tech (Regular-Full time)

(Effective for the students admitted into I year from the Academic Year **2026-27** onwards)

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE (AI&DS)

I & II YEARS COURSE STRUCTURE AND SYLLABUS

And

B.Tech(Lateral Entry Scheme)

(Effective for the students getting admitted into II year through Lateral Entry Scheme from the Academic Year 2027-28 onwards)

B.Tech. – I Year I Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AHS00104T	Physic for Advanced Computing	3	0	0	0	3
2	26AHS00102T	Linear Algebra and Calculus	3	0	2	0	4
3	26ACS05101T	Design Thinking	1	0	4	0	3
4	26ACS05102T	Computational Thinking and Problem Solving with Python	3	0	0	0	3
5	26ACS05103T	AI Tools and Applications	1	0	4	0	3
6	26ACS05102P	Computational Thinking and Problem Solving with Python Lab	0	0	0	3	1.5
7	26AHS00104P	Physic for Advanced Computing Lab	0	0	0	3	1.5
8	26AHS00101L	NSS/NCC/Scouts & Guides/Community Service	0	0	0	1	0.5
9	26ACS05101M	Cybersecurity Awareness for Engineers	2	0	0	0	0
		Total	**13**	**0**	**10**	**07**	**19.5**

I Year II Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	.26AHS00201T	Communicative English	2	0	0	0	2
2	26AHS00101T	Applied Chemistry for Computer Science	3	0	0	0	3
3	26AHS00203T	Differential Equations and Vector Calculus	3	0	2	0	4
4	26ACS05201T	**Professional Core - I** • Data Structures & Efficient Problem Solving	3	0	0	0	3
5	26AME03201T	Engineering Graphics	1	0	4	0	3
6	26AME03201P	Engineering Workshop	0	0	0	3	1.5
7	26AHS00101P	Applied Chemistry for Computer Science Lab	0	0	0	3	1.5
8	26AHS00201P	Communicative English Lab	0	0	0	2	1
9	26ACS05201P	Data Structures & Efficient Problem-Solving Lab	0	0	0	2	1
10	26AHS00201L	Health and wellness, Yoga and Sports	0	0	0	1	0.5
		Total	**12**	**0**	**6**	**13**	**20.5**

II Year I Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AHS00301T	Probability and Statistics	3	0	0	0	3
2	26AHS00304T	Biology for Engineers	2	0	0	0	2
3	26AEC04301T	Digital Logic and Computer Organization	3	0	0	0	3
4	26ACS05301T	**Professional Core - II** • Algorithms Design and Analysis	3	0	0	0	3
5	26ACS05302T	**Professional Core - III** • Object Oriented Design and Programming	3	0	0	0	3
6	26AEC04301P	Digital Logic and Computer Organization Lab	0	0	0	2	1
7	26ACS05301P	**Professional Core - II Lab** • Algorithms Design and Analysis Lab	0	0	0	3	1.5
8	26ACS05302P	**Professional Core - III Lab** • Object Oriented Design and Programming Lab	0	0	0	3	1.5
9	26AHS00305T	Sustainability and Green Engineering	2	0	0	0	2
10	26AAI30301S	**Skill Enhancement Course I** • AI Assisted Application Development	1	0	0	2	2
11	26AHS00302M	Technical Paper Writing & IPR	2	0	0	0	0
Total			**19**	**0**	**0**	**10**	**22**

II Year II Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AMB20301T	Managerial Economics and Financial Analysis	2	0	0	0	2
2	26AHS00302T	Universal Human Values – Understanding Harmony and Ethical Human Conduct	3	0	0	0	3
3	26AAI30401T	**Professional Core - IV** • Intelligent Data Preparation and Feature Engineering	3	0	0	0	3
4	26AAI30402T	**Professional Core - V** • Intelligent Database Management Systems	3	0	0	0	3
5	26AAI30403T	**Professional Core - VI** • Streaming Data Engineering and Pipelines	2	0	2	0	3
6	26AAI30404T	**Professional Core - VII** • Fundamentals of Artificial Intelligence	2	0	2	0	3
7	26AAI30401P	**Professional Core - IVLab** • Intelligent Data Preparation and Processing Systems Lab	0	0	0	3	1.5
8	26AAI30402P	**Professional Core - VLab** • Intelligent Database Management Systems Lab	0	0	0	3	1.5
9	26ACS05405S	**Skill Enhancement course –II** Data Analytics Using Python	1	0	0	2	2
10	26AHS00301M	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							

Course Code	PHYSICS FOR ADVANCED COMPUTING	L	T	Pr	P	Credits
26AHS00104T	Common to CSE & Allied Branches	3	0	0	0	3
Prerequisite:	Engineering Mathematics, Basic Electronics, Computer Science Fundamentals					

Prerequisites: Engineering Mathematics, Basic Electronics, Computer Science Fundamentals

Course Objectives

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

Course Outcomes (COs)

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

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)**

Course Content

UNIT – 1	Wave Optics and Quantum Information Processing	9 Hrs
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 AI Tools: Python (NumPy, SciPy, Matplotlib and scikit-learn).

UNIT – 2	**Quantum Photonics and Advanced Sensing**	**9 Hrs**
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 AI Tools: Google Colab, Python, NumPy, SciPy and Scikit-learn/TensorFlow.	

UNIT – 3	**Introduction to Quantum Mechanics and Quantum Computing**	**10 Hrs**
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 AI Tools: Google Colab, Python and Scikit-learn.	

UNIT – 4	**Semiconductors and Magnetic Materials**	**8 Hrs**
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 – 5	**Low Temperature Physics and Superconductivity**	**9 Hrs**

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.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L3	Apply	**CO1, CO3, CO5**	Interference and diffraction problems; quantum tunnelling application exercises; superconductivity application case studies.
L4	Analyze	**CO2, CO4**	Laser and fiber optic system analysis; Hall effect and magnetic material characterization exercises.

CO – PO Articulation Matrix (NBA)

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

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
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

CO-PO Correlation Highlights	CO1, CO3, and CO5 strongly align with PO1 (Engineering Knowledge) through application of optics, quantum mechanics, and superconductivity principles. CO2 and CO4 map to PO2 and PO5 through analysis of photonic systems and AI-assisted material characterization. CO5 also connects to PO7 (Environment & Sustainability) through cryogenic energy efficiency. All COs link to PO12 (Lifelong Learning) through continuous engagement with emerging AI-assisted physics analysis tools, and to PO6, PO8–PO10 at a slight level through engineering ethics, teamwork in lab settings, and reporting of experimental findings.

LEARNING RESOURCES

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

- NPTEL – nptel.ac.in
- MIT OpenCourseWare – ocw.mit.edu
- HyperPhysics – hyperphysics.phy-astr.gsu.edu
- PhET Simulations – phet.colorado.edu
- 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

Course Code	LINEAR ALGEBRA AND CALCULUS	L	T	Pr	P	Credits
26AHS00102T	AI & DS | I Year I Sem	3	0	2	0	4
Prerequisite:	Higher Secondary Mathematics					

Prerequisites: Higher Secondary Mathematics

Course Objectives

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

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)**

Course Content

UNIT – 1	Matrix Algebra and Linear Systems	9 Hrs
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; Jupyter Notebook; Scilab.

UNIT – 2	Eigenvalues, Eigenvectors, Diagonalization and Quadratic Forms	9 Hrs
Content	• Eigenvalues and eigenvectors; characteristic equation; Cayley-Hamilton theorem; 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	9 Hrs
Content	• Rolle's theorem; Lagrange's mean value theorem; Cauchy's mean value theorem; Taylor's and Maclaurin's series with remainder; 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; WolframAlpha free tier.	

UNIT – 4	Multivariable Differentiation and Optimization	9 Hrs
Content	• Functions of several variables; limits and continuity; partial derivatives; higher order partial derivatives; total derivative; chain rule; gradient; 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	9 Hrs
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.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L3	Apply	**CO1, CO3**	Gauss elimination and Gauss-Jordan problem-solving; Taylor series approximation exercises.
L4	Analyze	**CO2, CO4**	Eigenvalue and diagonalization analysis; Jacobian and constrained optimization problems.
L5	Evaluate	**CO5**	Multiple integral evaluation for area/volume/centroid computation; coordinate transformation assessment.

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	3	2	2	1	-	-	-	1	2	-	-
CO2	3	3	3	2	1	-	-	-	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 and CO2 strongly align with PO1 (Engineering Knowledge) and PO2 (Problem Analysis) through matrix and eigenvalue problem-solving. CO4 and CO5 map to PO3 and PO4 through optimization and multiple integral applications. PO5 (Modern Tools) is addressed throughout via Python, SymPy, and Google Colab integration. CO5 additionally links to PO7 and PO8 through engineering field interpretation, and strongly to PO12 (Lifelong Learning) through continued computational skill development beyond the course.

LEARNING RESOURCES

Textbooks

1. Grewal, B. S. (2017). Higher Engineering Mathematics (44th ed.). Khanna Publishers.
2. Kreyszig, E. (2018). Advanced Engineering Mathematics (10th ed.). Wiley.

Reference Books

1. Thomas, G. B., Weir, M. D., & Hass, J. (2018). Thomas' Calculus (14th ed.). Pearson.

2. Jain, R. K., & Iyengar, S. R. K. (2021). Advanced Engineering Mathematics (5th ed.). Alpha Science International.
3. Strang, G. (2016). Introduction to Linear Algebra (5th ed.). Wellesley-Cambridge Press.
4. Lay, D. C., Lay, S. R., & McDonald, J. J. (2015). Linear Algebra and Its Applications (5th ed.). Pearson.
5. Arfken, G. B., Weber, H. J., & Harris, F. E. (2012). Mathematical Methods for Physicists (7th ed.). Academic Press.
6. Zill, D. G., & Wright, W. S. (2018). Advanced Engineering Mathematics (6th ed.). Jones & Bartlett.

Online Resources

- NPTEL: Linear Algebra – https://nptel.ac.in
- MIT OpenCourseWare – Linear Algebra: https://ocw.mit.edu
- MIT OpenCourseWare – Multivariable Calculus: https://ocw.mit.edu
- Python SymPy Documentation: https://docs.sympy.org
  - Google Colab for Python: https://colab.research.google.com

Recommended Free & Open-Source AI/Computational Tools

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

Course Code	DESIGN THINKING	L	T	Pr	P	Credits
26ACS05101T	AI & DS | I Year I Sem	1	0	4	0	3
Prerequisite:	None					

Prerequisites: None

Course Objectives

- Introduce the philosophy, principles, and five-stage model of Design Thinking as a human-centred innovation methodology.
- Develop empathy-driven research skills including interviews, observation, and persona creation to understand user needs.
- Apply structured ideation techniques and AI-assisted tools to generate and prioritise creative solutions.
- Build low-fidelity and digital prototypes incorporating AI-assisted design tools.
- Conduct user testing, iterate based on feedback, and present design solutions effectively.

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 philosophy, principles, and stages of the Design Thinking process and its relevance to human-centred problem solving.	**L1–L2 (Remember/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)**

Course Content

UNIT – 1	Introduction to Design Thinking	9 Hrs
Content	• Design Thinking: Definition, origin (Stanford d.school, IDEO), and relevance to engineering. • The 5-Stage Model: Empathise – Define – Ideate – Prototype – Test. • Human-centred design principles: Desirability, Feasibility, Viability. • Systems Thinking vs. Linear Thinking: Understanding complexity. • Case Studies: Successful design thinking applications (IDEO shopping cart, Stanford Mobile Clinic).	
AI	AI Tool: ChatGPT/Copilot – use AI to explain design thinking in different domains	

Integration	(healthcare, education, agriculture). Activity: MindMeister (free, open-source alternative: FreeMind) to create a mindmap of the 5-stage process. AI Video Analysis: summarise key DT principles from documentary case studies.

UNIT – 2	**Empathise and Define Stages**	**9 Hrs**
Content	• Empathy Methods: Interviews, Observation, Immersion, Shadowing. • Creating User Personas: Demographic, psychographic, goals and pain points. • Empathy Mapping: Says, Thinks, Does, Feels. • Point of View (POV) Statement: User + Need + Insight formula. • How Might We (HMW) Questions: Reframing problems as opportunities.	
AI Integration	AI Tool: Miro (free plan) for digital empathy maps and POV statements. AI Tool: ChatGPT for generating diverse user persona profiles. Activity: interview peers about a campus problem and use AI to analyse transcripts and extract themes. AI Tool: Dovetail (free plan) for affinity mapping of qualitative research data.	

UNIT – 3	**Ideation Stage**	**9 Hrs**
Content	• Creative Thinking Techniques: Brainstorming, Brainwriting, SCAMPER, Random Word Association. • 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.	
AI Integration	AI Tool: ChatGPT – use 'expand my idea' prompts to generate variations of a concept. AI Tool: IdeaFlip (free)/Miro for digital brainstorming and affinity diagramming. AI Tool: Midjourney (free trial)/DALL-E (free credits) for visual concept sketches. Activity: run an AI-augmented brainstorming session and compare idea quality with pure human brainstorming.	

UNIT – 4	**Prototyping**	**9 Hrs**
Content	• Principles of Prototyping: Low-fidelity vs. High-fidelity prototypes. • Paper Prototyping: Sketching UI and physical models. • Digital Wireframing: Introduction to wireframes and mockups. • 3D Prototyping Concepts: Introduction to 3D printing and rapid prototyping. • Role Plays and Experience Prototyping: Service design scenarios.	
AI Integration	AI Tool: Figma AI for converting sketches to digital wireframes. AI Tool: Tinkercad (free, web-based) for simple 3D models for physical prototyping. Activity: build a paper prototype for a chosen problem and photograph for digital documentation.	

UNIT – 5	**Testing, Iteration and AI in Design**	**9 Hrs**
Content	• User Testing Methods: Think-aloud protocol, A/B testing, Usability testing. • Feedback Analysis: Affinity mapping of test 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).	

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 final prototype to a panel; use AI to prepare pitch slides in Canva. Reflection: Design Thinking journal documenting each stage with AI tool usage notes.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1/L2	Remember/Understand	**CO1**	Design Thinking principle identification; 5-stage model recall exercises.
L3	Apply	**CO2**	Empathy interview conduction; persona creation and POV statement formulation.
L4	Analyse	**CO3**	HMW framework problem reframing; AI-assisted insight generation analysis.
L5/L6	Evaluate/Create	**CO4, CO5**	Prototype evaluation using testing rubrics; final prototype creation and stakeholder pitch.

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
Average	1.6	2.6	2.8	2.2	2.4	2.4	1.2	1.4	2.4	2.2	2.0	2.6	2.0	2.0

Text Books

1. T1. Brown, Tim. (2019). Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation. HarperBusiness, New York.
2. T2. Lewrick, Michael, Link, Patrick & Leifer, Larry. (2018). The Design Thinking Playbook. Wiley, New Jersey.

Reference Books

1. R1. IDEO.org. (2015). The Field Guide to Human-Centered Design. IDEO.org, San Francisco.
2. R2. Kelley, Tom & Kelley, David. (2013). Creative Confidence: Unleashing the Creative Potential Within Us All. Crown Business.
3. R3. Liedtka, Jeanne & Ogilvie, Tim. (2011). Designing for Growth: A Design Thinking Tool Kit for Managers. Columbia Business School Press.
4. R4. Plattner, Hasso, Meinel, Christoph & Leifer, Larry (Eds.). (2018). Design Thinking Research. Springer.

5. R5. Kumar, Vijay. (2012). 101 Design Methods: A Structured Approach for Driving Innovation. Wiley.
6. R6. Stickdorn, Marc & Schneider, Jakob. (2021). This Is Service Design Thinking (Expanded ed.). BIS Publishers.

Web Resources

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

YouTube / Video Resources

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

I Year I Sem

Course Code	COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON	L	T	Pr	P	Credits
26ACS05102T	Common to All Branches	3	0	0	0	3

Course Objectives

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

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)	L3 (Apply)
CO2	Develop programs using decision-making and looping constructs to solve logical problems. (L3)	L3 (Apply)
CO3	Analyse and manipulate data using strings, lists, tuples, and sets for problem solving. (L4)	L4 (Analyse)
CO4	Design modular programs using functions, recursion, and dictionaries to improve code reusability. (L4)	L4 (Analyse)
CO5	Implement file handling, exception handling, and basic object-oriented programming concepts in Python applications. (L3)	L3 (Apply)

Course Content

UNIT – I	COMPUTATIONAL THINKING AND PROGRAMMING BASICS	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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.

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 PrjMgm
CO1	L3	3	2	1	1	1	1	–	–	–	1	–
CO2	L3	3	3	2	1	1	–	–	–	–	1	–
CO3	L4	3	3	2	2	2	–	–	–	–	1	–
CO4	L4	3	3	3	2	2	–	–	–	1	2	–
CO5	L3	3	3	2	1	2	–	–	–	–	1	–

CO-PO Correlation	Strongest correlation with PO1 (Engineering Knowledge) through Python programming fundamentals and PO2 (Problem Analysis) through algorithm

Highlights	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.

LEARNING RESOURCES

Textbooks

1. TB1: Pieter Spronck, Computational Thinking with Python, (course textbook — covers computational thinking, algorithms, and Python programming).
2. TB2: Reema Thareja, Programming in Python, Oxford University Press. (Primary implementation reference — all five units.)

Reference Books

1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly Media. (Free at greenteapress.com)
2. Eric Matthes, Python Crash Course, 2nd Edition, No Starch Press.

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

I Year I Sem

Course Code	AI TOOLS AND APPLICATIONS	L	T	Pr	P	Credits
26ACS05103T	Common to All Branches	1	0	4	0	3

Course Objectives

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

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)

Course Content

UNIT – I	UNDERSTANDING AI AND AI FOR DOCUMENTS AND PRESENTATIONS	9 Hrs

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**	**9 Hrs**

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	9 Hrs

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	9 Hrs

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	9 Hrs

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.

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.

LEARNING RESOURCES

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.
4. Google – Fundamentals of AI – grow.google/intl/en_in/ai-learning-resources
5. Microsoft – AI Skills Navigator – microsoft.com/en-us/ai/ai-skills
6. Andrew Ng – AI for Everyone (Coursera, free to audit) – coursera.org/learn/ai-for-everyone
7. Elements of AI – University of Helsinki (free, beginner) – elementsofai.com
8. AI Assistants: ChatGPT – chat.openai.com | Claude – claude.ai | Gemini – gemini.google.com | Microsoft Copilot – copilot.microsoft.com
9. Research & Notes: NotebookLM – notebooklm.google.com | Perplexity AI – perplexity.ai | Elicit – elicit.org | Consensus – consensus.app
10. Presentations & Docs: Gamma.app | Notion AI – notion.so | Beautiful.ai | Copilot in Office 365 | Gemini in Google Workspace
11. Diagrams & Design: Napkin.ai | Whimsical AI – whimsical.com | Miro AI – miro.com | Canva AI – canva.com | DALL-E 3 via ChatGPT
12. Data & Spreadsheets: ChatGPT Advanced Data Analysis | Julius AI – julius.ai | Copilot in Excel | Gemini in Sheets
13. Math & Equations: Wolfram Alpha – wolframalpha.com | Symbolab – symbolab.com | Mathway – mathway.com | Microsoft Math Solver – math.microsoft.com | Photomath – photomath.com | Desmos – desmos.com | GeoGebra – geogebra.org
14. Audio & Video: ElevenLabs – elevenlabs.io | Suno – suno.com | Runway – runwayml.com | HeyGen – heygen.com
15. Detection & Automation: GPTZero – gptzero.me | Originality.ai | Google Lens | Make.com | GitHub Copilot

Online Resources

- https://www.youtube.com/@JoyofLearningCSE/featured
- EU AI Act – eur-lex.europa.eu | UNESCO AI Ethics – unesco.org | NITI Aayog – niti.gov.in

India DPDP Act 2023 – meity.gov.in | ProPublica COMPAS – propublica.org | MIT Technology Review – technologyreview.com |

I Year I Sem

Course Code	COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON LAB	L	T	Pr	P	Credits
26ACS05102P	Common to All Branches	0	0	0	3	1.5

Course Objectives

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

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 — decomposition, abstraction, pattern recognition — to design algorithms, flowcharts, and Python programs. (L3)	L3 (Apply)
CO2	Develop programs using decision-making and looping constructs to solve logical and iterative problems. (L3)	L3 (Apply)
CO3	Analyse and manipulate data using strings, lists, tuples, and sets for practical problem solving. (L4)	L4 (Analyse)
CO4	Design modular programs using user-defined functions, recursion, lambda expressions, and dictionaries. (L4)	L4 (Analyse)
CO5	Implement file handling, exception handling, and basic object-oriented concepts in Python applications. (L3)	L3 (Apply)

List of Experiments

Unit I	Computational Thinking and Programming Basics

1. **Problem Solving Using Computational Thinking**
   Select three daily-life problems (e.g., making tea, finding the shortest route to college, calculating a monthly budget). Apply decomposition (break into sub-problems), abstraction (identify essential information), and pattern recognition to each. Write a step-by-step algorithm for each problem; convert one algorithm into pseudocode. Use Python Tutor or draw.io to trace and visualise the solution flow; use ChatGPT to generate an alternate solution and compare both approaches for efficiency.

2. **Flowchart Design and Algorithm Simulation**
   Design flowcharts using Flowgorithm or draw.io for three problems: (a) find the largest of three numbers; (b) compute a student's grade (O/A+/A/B+/B/C/F) based on percentage; (c) generate an electricity bill (slab-based tariff). Trace each flowchart manually with 3 sample inputs before coding. Simulate

execution in Flowgorithm and verify that the output matches the manual trace. Use ChatGPT or Algorithm Visualizer to identify logic errors in any incorrect flowchart and document the correction.

3. Python Programming Fundamentals

Write Python programs to: (a) declare variables of all data types (int, float, complex, bool, str); print each with its type using type(); (b) demonstrate all operator categories — arithmetic, relational, logical, bitwise, assignment — with one example each; (c) implement three application programs: Simple Interest (P, R, T as input), Area of Circle/Rectangle/Triangle (menu-driven), Unit conversion (km to miles, Celsius to Fahrenheit, kg to pounds); (d) demonstrate implicit and explicit type conversion — int to float, float to int, str to int using int(), str() — and print a formatted bill using f-strings. Use ChatGPT to generate 5 test inputs and verify program output against each.

Unit II | Decision Making and Looping

4. Decision Control Statements

Develop Python programs for: (a) find the largest among three numbers using if-elif-else; (b) grade calculation — accept percentage and print grade (O ≥ 90, A+ ≥ 80, A ≥ 70, B+ ≥ 60, B ≥ 50, C ≥ 40, F < 40); (c) leap year check using nested if — handle both century and non-century years; (d) menu-driven calculator — display 5 operations (+, −, ×, ÷, %); accept operator and two numbers; handle division by zero. Use ChatGPT to generate boundary test cases (e.g., exactly 90%, year 2000, year 1900) and test each program against them before submission.

5. Looping and Iterative Problem Solving

Implement Python programs for: (a) print all prime numbers between 1 and 100 using a while loop with a flag variable; also count and display the total; (b) accept a number and check if it is an Armstrong number (e.g., $153 = 1^3 + 5^3 + 3^3$); find all Armstrong numbers between 1 and 1000; (c) compute factorial of a number using a for loop; also compute factorial recursively — compare both outputs; (d) print the multiplication table of a number entered by the user (1 to 20) in a formatted table using a for loop. Use Python Tutor to trace loop execution for the Armstrong number check with input 153; verify the iteration count.

6. Nested Loops and Pattern Generation

Generate the following patterns using nested for loops — accept the number of rows (n) as input for each: (a) right-angled triangle of stars, inverted triangle of stars; (b) number pyramid (1, 1 2, 1 2 3…), Floyd's triangle (consecutive numbers); (c) diamond pattern (upper triangle + lower triangle); (d) alphabet pattern (A, AB, ABC… and A, BC, DEF…). For each pattern, first trace the row and column values manually in a table before coding. Use ChatGPT to explain the iteration flow for one pattern; compare the AI explanation with your own trace and document any differences.

Unit III | Strings and Data Structures

7. String Processing Applications

Develop Python programs for: (a) accept a string and: reverse it, check palindrome, count vowels/consonants/digits/special characters, count words, convert to title case; (b) find frequency of each character (store in dictionary), print characters sorted by frequency descending; (c) accept a paragraph and: find the longest word, replace all occurrences of a given word, print words that appear more than once; (d) implement a Caesar cipher — encrypt a message by shifting each letter by a given value (1–25); write a decrypt function to recover the original. Use ChatGPT to generate 5 string test cases with edge conditions (empty string, all spaces, mixed case, special characters) and test each program.

8. List and Nested List Operations

Perform the following list operations: (a) create a list of 10 integers; implement insert at a position, delete by value, find maximum and minimum without built-in functions, sort using Bubble Sort, and linear search — display the list after each operation; (b) demonstrate list slicing (first half, second half, alternate

elements, reverse using slice) and list comprehension (squares of even numbers, words longer than 4 characters); (c) implement matrix operations using nested lists: addition of two 3×3 matrices, transpose of a matrix, multiply two 2×2 matrices; display results in matrix format; (d) store student records as a list of tuples (roll_no, name, marks); sort by marks descending using sorted() with a lambda key; print a formatted rank list. Use ChatGPT to suggest optimisations and compare with your implementation.

9. Tuple and Set Applications

Implement the following programs: (a) demonstrate tuple packing and unpacking — pack student data (roll_no, name, marks) into a tuple; unpack and display each field; store 5 student records as a list of tuples and display in tabular format; (b) demonstrate all set operations on two user-defined sets A and B: union (A | B), intersection (A & B), difference (A − B, B − A), symmetric difference (A ^ B); verify using membership testing (in, not in); (c) find students who passed all three subjects, students who failed at least one, and students who passed exactly two — represent each group as a set and use set operations; (d) demonstrate frozenset — create a frozenset of day names; show that it cannot be modified (TypeError); use it as a dictionary key. Use ChatGPT to explain why sets and frozensets cannot contain mutable elements.

Unit IV | Functions and Problem Solving

10. User-Defined Functions

Implement Python programs demonstrating all four argument types: (a) positional — def calculate_bmi(weight, height): return weight/height**2; call with 3 different inputs; (b) default — def greet(name, msg='Hello'): print(msg, name); call with and without the default; (c) keyword — def student_info(name, roll, marks): call using greet(marks=85, name='Arun', roll=101); (d) variable-length — *args: def total(*numbers): return sum(numbers); **kwargs: def display_profile(**details): print each key-value pair. Write 5 utility functions (is_prime, is_palindrome, factorial, gcd, power) and call each with 3 test inputs. Use ChatGPT to generate additional test cases and verify all function outputs.

11. Recursive Problem Solving

Develop recursive solutions for: (a) factorial(n) and fibonacci(n) — implement both recursive and iterative versions; measure execution time using time.time() for n = 5, 10, 15, 20; report crossover point where recursion becomes slower; (b) GCD of two numbers using Euclid's algorithm recursively — verify using math.gcd(); (c) Tower of Hanoi for n = 3 disks — print each move (disk, from peg, to peg); count total moves and verify it equals $2^n - 1$; (d) binary search recursively — implement binary_search(arr, target, low, high); test on a sorted list of 20 elements with 5 targets (2 present, 3 absent). Use Python Tutor to trace the recursion tree for factorial(5) and Fibonacci(6); count the total recursive calls for each.

12. Dictionary and Lambda Applications

Implement the following: (a) student database using a dictionary {roll_no: {name, marks_list}}; add, update, delete, and search students; compute average marks and find topper; display all records sorted by average marks descending; (b) word frequency counter — accept a paragraph; build a {word: count} dictionary; print top 5 most frequent words; (c) demonstrate all dictionary methods: keys(), values(), items(), get(), update(), pop(), setdefault(), copy() — with one practical example each; (d) lambda applications — define lambda functions for: square, cube, even/odd filter, Celsius-to-Fahrenheit; use each with map(), filter(), and sorted() on a list of 10 elements; demonstrate that lambda works as an argument to a higher-order function.

Unit V | File Handling, Exception Handling, and OOP

13. File Processing and Data Storage

Develop Python programs to: (a) create a text file and write 10 student records (roll number, name, marks); read and display all lines; count total words and lines; append 5 more records; display only records where marks > 60; (b) copy contents of one file to another; replace all occurrences of a specific word (e.g., 'Python' → 'programming') and write to a new file; (c) read and process a CSV file (student

data) using the csv module — compute average marks per subject, find the highest and lowest scorers, write a summary CSV with name and grade; (d) demonstrate all four file modes ('r', 'w', 'a', 'r+') with one practical example each. Use ChatGPT to generate a 20-row sample CSV dataset for testing.

14. **Exception Handling and Testing**
Implement exception handling programs: (a) division program — accept two numbers; handle ZeroDivisionError, ValueError (non-numeric input), and TypeError in separate except blocks; use finally to print 'Operation complete' regardless; (b) file access program — handle FileNotFoundError and PermissionError; print a user-friendly message (not a traceback) for each; (c) define a custom exception class InsufficientFundsError(Exception) with a message attribute; implement a BankAccount class with deposit() and withdraw() — withdraw() raises InsufficientFundsError when balance is insufficient; test with 5 deposit/withdrawal sequences; (d) implement input validation using exception handling — accept age (must be 1–120), email (must contain @), and phone (must be 10 digits); raise ValueError with a descriptive message for each invalid input. Use ChatGPT to generate 5 error scenarios per program and test.

15. **Object-Oriented Programming — Mini Project**
Design and implement a Python OOP mini project (choose one: Student Management System, Library Management, or Banking System). The project must include: (a) at least one class with minimum 5 attributes, a constructor (__init__), and 5 methods including one class method (using @classmethod) and one static method (using @staticmethod); (b) a class variable that tracks the total number of objects created; (c) object interactions — at least two classes with one referencing the other (e.g., Library has-a BookCollection); (d) file persistence — write all records to a text or CSV file on exit and read them back on program start. Use ChatGPT to generate a UML class diagram for the chosen system and use draw.io to draw it; implement the design and verify that the final program matches the diagram.

AI Tools Integration (Lab)	Use Thonny IDE (beginner-friendly debugger, variable inspector) or Google Colab as the primary lab environment. Use Python Tutor (pythontutor.com) to visualise variable states, call stacks, and list/dictionary memory — mandatory for Experiments 3, 11, and 15. Use Flowgorithm or draw.io to design flowcharts before writing code in Experiments 1 and 2. Use ChatGPT, Claude, or Microsoft Copilot to explain error messages — understand the explanation, fix the error independently, and document both in your lab record. Do NOT copy AI-generated code directly; use AI to understand, then write the solution yourself.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L3	Apply	**CO1–CO5**	Execute Python programs for algorithms, loops, string operations, and file handling.
L4	Analyse	**CO3, CO4**	Compare list vs. set performance; analyse recursive execution using Python Tutor.
L5	Evaluate	**CO5**	Evaluate exception handling strategies in an OOP application.
L6	Create	**CO4, CO5**	Build a Student Management System using OOP with file persistence.

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	1	1	1	1	–	–	–	1	–
CO2	L3	3	3	2	1	1	–	–	–	–	1	–
CO3	L4	3	3	2	2	2	–	–	–	1	1	–
CO4	L4	3	3	3	2	2	–	–	–	1	2	–
CO5	L3	3	3	2	1	2	–	–	–	–	1	–

CO-PO Correlation Highlights	Lab sessions reinforce PO1 (Engineering Knowledge) and PO2 (Problem Analysis) through hands-on algorithm implementation. PO3 (Design/Development) through OOP class design in Experiment 15. PO5 (Modern Tools) through Thonny, Python Tutor, Flowgorithm, and Google Colab across all experiments.

LEARNING RESOURCES

Textbooks

1. TB1: Pieter Spronck, Computational Thinking with Python (course textbook — algorithms, flowcharts, and Python foundations).
2. TB2: Reema Thareja, Programming in Python, Oxford University Press.

Reference Books

1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly. (Free at greenteapress.com)
2. Eric Matthes, Python Crash Course, 3rd Edition, No Starch Press.
3. John Zelle, Python Programming: An Introduction to Computer Science, Franklin Beedle.

Online Resources

- ChatGPT – chat.openai.com | Microsoft Copilot – copilot.microsoft.com | Claude – claude.ai
- Flowgorithm – flowgorithm.org |Lucidchart AI – lucidchart.com | draw.io – draw.io
- Python Tutor – pythontutor.com | Algorithm Visualizer – algorithm-visualizer.org
- Thonny IDE – thonny.org | Google Colab – colab.research.google.com
- NPTEL – Problem Solving through Programming in Python – nptel.ac.in
- W3Schools Python – w3schools.com/python | Python Official Docs – docs.python.org/3
-

Course Code	PHYSICS FOR ADVANCED COMPUTING LAB	L	T	Pr	P	Credits
26AHS00104P	Common to CSE & Allied	0	0	0	3	1.5

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)

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

Web Resources:

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

Recommended AI Tools:

- Python (NumPy, Pandas, Matplotlib)
- TensorFlow / Scikit-learn

Course Code	NSS / NCC / Scouts & Guides / Community Service	L	T	Pr	P	Credits
26AHS00101L	Common to All Branches	0	0	0	1	0.5

Course Objectives

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

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)

Unit-wise Syllabus with Learning Outcomes

UNIT 1	Foundation of NSS, NCC, Scouts & Guides and Community Service	5 Hrs
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	5 Hrs
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 (ULOs)	**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] • 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	5 Hrs
Topics	• Community needs assessment – participatory methodology • Project planning: Swachh Bharat, Pulse Polio, Blood Donation camps • Literacy campaigns and awareness drives (PadhnaLikhna 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]	

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

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

YouTube Resources

- NSS Song & Pledge (Official MYA&S) – https://youtu.be/nss_official_01
- NCC Drill & Parade Training Basics – https://youtu.be/ncc_drill_02
- Community Service Project Planning – https://youtu.be/community_svc_03
- First Aid & CPR Tutorial – Red Cross India – https://youtu.be/firstaid_rci_04
- Disaster Management Awareness – NDMA – https://youtu.be/ndma_dm_05
- Baden-Powell and the Scout Movement – https://youtu.be/scouts_bp_06
- Leadership Skills for Youth – https://youtu.be/leadership_youth_07
- Swachh Bharat Campaign Activities – https://youtu.be/swachh_sb_08

AI-Integrated Activities

S.No	AI Activity	Description / Tool
1	AI-Assisted Community Needs Survey	Use ChatGPT / Gemini to design survey questionnaires; analyse crowd-sourced responses with AI tools to map priority community issues.
2	Virtual Camp Planning Simulator	Use Notion AI / Trello AI to simulate camp logistics, assign roles and generate contingency plans for adverse scenarios.
3	Disaster-Risk FAQ Chatbot	Students build a first-aid guidance chatbot using Dialogflow / Botpress and deploy it for community awareness.
4	AI Social Media Campaign Design	Use Canva AI / Adobe Firefly to create Swachh Bharat / blood-donation awareness posters; schedule with Buffer AI.
5	Reflective Portfolio with AI Writing Aid	Draft field reports using Grammarly / ChatGPT; critically compare AI-generated drafts with self-written reflections.

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	-	-	-	-	-

I Year II Sem

Course Code	CYBERSECURITY AWARENESS FOR ENGINEERS	L	T	Pr	P	Credits
26ACS05101M	AI & DS | I Year II Sem	2	0	0	0	0
Prerequisite:	None					

Prerequisites: None

Course Objectives

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

Course Outcomes (COs)

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

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–L3**
CO5	Develop cyber hygiene habits and respond appropriately to common cybersecurity incidents.	**L3 (Apply)**

Course Content

UNIT – 1	Introduction to Cyberspace and Cybersecurity	6 Hrs
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 – 2	Cyber Threats, Cybercrime and Social Engineering	6 Hrs
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 – 3	Personal Cyber Safety and Cybersecurity Technologies	6 Hrs
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 – 4	Privacy, Cyber Ethics and Cyber Laws	6 Hrs
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 – 5	Cyber Hygiene and Secure Digital Practices	6 Hrs
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.	

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L2	Understand	**CO1, CO2, CO4**	CIA Triad identification; malware type classification; cyber law awareness quiz.
L3	Apply	**CO3, CO5**	MFA setup exercises; secure browsing practice tasks; incident response drill.

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	2	1	1	–	–	2	–	2	–	1	–	2
CO2	2	2	1	–	–	2	–	2	–	1	–	2
CO3	1	1	2	–	2	2	–	2	–	1	–	2
CO4	1	1	1	–	–	2	–	3	–	2	–	2
CO5	1	1	2	–	1	3	–	2	1	1	–	2

CO-PO Correlation Highlights	CO5 shows the strongest alignment with PO6 (Engineer & Society) reflecting the course's focus on responsible digital citizenship. CO4 maps strongly to PO8 (Ethics) through cyber law and privacy awareness. As a 0-credit mandatory course, the emphasis is on PO6, PO8, and PO12 (Lifelong Learning) rather than technical PO1-PO5 correlations.

LEARNING RESOURCES

Textbooks

1. Singh, A. (n.d.). Introduction to Cyber Security: Concepts, Principles, Technologies and Practices. Universities Press / Orient BlackSwan.
2. Godbole, N., & Belapure, S. (n.d.). Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives. Wiley India.

Reference Books

1. Brooks, C. J., Grow, C., Craig, P., & Short, D. (n.d.). Cybersecurity Essentials. Wiley.
2. Stallings, W., & Brown, L. (n.d.). Computer Security: Principles and Practice. Pearson.
3. Whitman, M. E., &Mattord, H. J. (n.d.). Principles of Information Security. Cengage Learning.
4. Kizza, J. M. (n.d.). Guide to Computer Network Security. Springer.
5. Cole, E. (n.d.). Cybersecurity for Beginners. Wiley.

Online Resources

- National Cyber Security Awareness Portal (India)
- CERT-In (Indian Computer Emergency Response Team)
- National Cyber Crime Reporting Portal

- NIST Cybersecurity Framework Resources
- OWASP Foundation

Cisco Networking Academy – Introduction to Cybersecurity

I Year II Sem

Course Code	COMMUNICATIVE ENGLISH	L	T	Pr	P	Credits
26AHS00201T	Common to All Branches	2	0	0	0	2

Course Objectives

Code	Bloom's Level	Statement
CO1	Remember	Recall and list fundamental concepts of English grammar, vocabulary, phonetics, and communication models essential for professional contexts.
CO2	Understand	Explain and interpret various forms of written and oral communication including emails, reports, presentations, and group discussions.
CO3	Apply	Use appropriate AI tools and digital platforms to enhance communicative accuracy, fluency, and professional writing skills in real-world scenarios.
CO4	Analyze	Examine and differentiate effective versus ineffective communication strategies in professional, academic, and intercultural contexts.
CO5	Evaluate	Appraise and critique written and spoken communication samples using AI-assisted feedback mechanisms and peer-review methods.
CO6	Create	Design and produce professional-grade communication artifacts — including portfolios, reports, and presentations — integrating AI tools for quality enhancement.

Course Outcomes (COs)

Upon successful completion, students will demonstrate the following outcomes measured through formative and summative assessments, mapped to Bloom's Taxonomy:

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

Detailed Syllabus

UNIT I: Foundations of English Communication

Weeks 1–3 | 6 Contact Hours

Topics Covered

- Basics of Communication: Process, Types (Verbal/Non-verbal), and Barriers
- Parts of Speech: Nouns, Pronouns, Verbs, Adjectives, Adverbs
- Tenses: Simple, Continuous, Perfect, and Perfect Continuous
- Subject-Verb Agreement and Correction of Sentences

Learning Outcomes

1. Recall the fundamental components and types of the communication process. (Remember – CO1)
2. Identify and classify parts of speech and tense forms in given sentences. (Remember – CO1)
3. Explain sentence structure patterns and apply them in original writing samples. (Understand – CO2)
4. Articulate words with correct stress and intonation using digital phonetics tools. (Apply – CO3)

AI Tools Integrated

Tool 1	ChatGPT (OpenAI)	Grammar explanation, error identification, and instant Q&A on language rules
Tool 2	Grammarly	Real-time grammar, spelling, and punctuation correction with detailed explanations
Tool 3	Google Speech-to-Text	Phonetics practice – voice recording for pronunciation and intonation feedback
Tool 4	Merriam-Webster Vocabulary Builder	Word-of-the-day, etymology, and interactive vocabulary quizzes

Hands-On AI Activity

Students paste a 100-word paragraph into Grammarly and use ChatGPT to understand each flagged error. They then record their voice reading the corrected paragraph using Google Speech-to-Text, receive pronunciation analysis, and submit: (a) original paragraph, (b) AI-corrected version with error report, (c) reflection note on errors identified.

UNIT II: Listening and Speaking Skills

Weeks 10–12 | 6 Contact Hours

Topics Covered

- Principles of Effective Speaking and Active Listening
- Group Discussion (GD): Techniques, Etiquette, and Evaluation Criteria
- Presentation Skills: Design Principles, Slide Preparation, and Delivery Techniques
- Debate: Argument Construction, Evidence Use, and Rebuttal Techniques

Learning Outcomes

1. Demonstrate effective speaking skills in structured group discussions and mock interview settings. (Apply – CO3)
2. Differentiate between effective and ineffective presentation strategies using AI-generated feedback. (Analyze – CO4)
3. Construct and deliver a well-organized 5-minute presentation using AI-designed slides. (Create – CO6)
4. Evaluate peer presentations using AI-powered transcription tools and structured assessment rubrics. (Evaluate – CO5)

AI Tools Integrated

Tool 1	Gamma.app / Beautiful.ai	AI-powered presentation design – structured, visual, professional slides
Tool 2	Orai (AI Speech Coach)	Fluency analysis, filler-word detection, pacing, and confidence scoring
Tool 3	Otter.ai	Auto-transcription and keyword analysis of spoken communication
Tool 4	Canva AI	Visual communication design – infographics, posters, and slide aesthetics

Hands-On AI Activity

Students prepare a 5-minute presentation on a technical or social topic using Gamma.app or Canva AI. Before class delivery, they practice using Orai (AI Speech Coach) to monitor pacing, filler words, and confidence score (minimum 3 practice sessions logged). Presentations are delivered in class, transcribed via Otter.ai, and peer-evaluated on rubric. Each student submits: (a) slides link, (b) Orai session screenshots, (c) Otter.ai transcript, (d) peer evaluation sheet.

UNIT III: Reading Comprehension & Writing Skills

Weeks 4–6 | 6 Contact Hours

Topics Covered

- Reading Strategies: Skimming, Scanning, Intensive and Extensive Reading
- Comprehension Passages, Synonyms, Antonyms, Words Often Confused, Homonyms, Homophones and Homographs
- Paragraph Writing: Topic Sentence, Supporting Details, Clincher Sentence
- Creative Writing – Features and Techniques – Movie Reviews & Blogs

Unit Learning Outcomes

1. Demonstrate skimming and scanning strategies to extract key information from texts. (Apply – CO3)
2. Summarize and paraphrase academic passages using AI paraphrasing tools accurately. (Understand – CO2)
3. Compose well-structured paragraphs and short essays on technical and social topics. (Apply – CO3)
4. Evaluate the clarity and readability of own writing using AI-based readability scoring tools. (Evaluate – CO5)

AI Tools Integrated

Tool 1	QuillBot	AI paraphrasing tool – rewrite passages at varying complexity levels
Tool 2	Hemingway Editor	Readability scoring, passive voice detection, and sentence simplification
Tool 3	ChatGPT	Essay outline generation, argument structuring, and content brainstorming
Tool 4	Rewordify	Vocabulary simplification for improving comprehension of complex texts

Hands-On AI Activity

Students upload a 300-word passage to QuillBot and generate three paraphrased versions (simple, standard, creative modes). They paste their own essay into Hemingway Editor and revise until readability reaches Grade 8 or below. Final submission includes: (a) original passage, (b) three paraphrased versions with mode labels, (c) before-and-after Hemingway screenshots with readability score comparison.

UNIT IV: Professional & Business Communication

Weeks 7–9 | 6 Contact Hours

Topics Covered

- Email Writing and Letter Writing: Formal & informal
- Short Report Writing: Structure, Types (Formal/Informal), and Format
- Note-Making and Summarization Techniques
- Resume / CV Writing and LinkedIn Profile Optimization

Learning Outcomes

1. Draft formal and semi-formal business emails and letters using professional writing conventions. (Apply – CO3)
2. Produce structured reports, proposals, and technical documents for workplace contexts. (Apply – CO3)
3. Design professional resumes and LinkedIn profiles using AI-assisted content generation tools. (Create – CO6)
4. Analyze the structure, register, and tone of professional documents for accuracy and appropriateness. (Analyze – CO4)

AI Tools Integrated

Tool 1	ChatGPT	Draft and refine professional emails, cover letters, reports, and proposals
Tool 2	Kickresume / Resume.io	AI-powered resume builder with ATS-optimized templates
Tool 3	Notion AI	Report structuring, agenda writing, and knowledge management
Tool 4	Grammarly Business	Tone detection, formality check, and professional register evaluation

Hands-On AI Activity

Each student drafts a formal job application email and a one-page resume using ChatGPT and Kickresume. They run both documents through Grammarly Business for tone and professionalism checks, then conduct peer evaluation using a structured rubric (structure, tone, grammar, format). Final submission: (a) AI-drafted email + resume, (b) Grammarly tone/score report, (c) peer feedback form.

UNIT V: Use of AI in Language Learning

Weeks 13–16 | 8 Contact Hours

Topics Covered

- Artificial Intelligence in Communication, Meaning and basics of AI & Benefits and limitations of AI in human interaction
- 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.
- 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)
- 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

Unit Learning Outcomes

1. Students understand the role of AI in modern communication and develop awareness of technology-driven language use. (CO1)
2. Students learn to use AI responsibly for improving speaking, writing, reading, and listening skills. (Evaluate – CO5)
3. Students develop ethical awareness, critical thinking, and responsible digital communication habits. (Create – CO6)
4. Students improve professional communication and employability skills using modern AI tools. (Create – CO6)

AI Tools Integrated

Tool 1	Copyleaks / Turnitin	Plagiarism detection, paraphrase checker, and academic integrity analysis
Tool 2	DeepL Translator	High-accuracy translation supporting intercultural communication understanding
Tool 3	LinkedIn Learning / Coursera	AI literacy modules – understanding AI bias, ethics, and responsible usage
Tool 4	Wix / Adobe Portfolio	E-portfolio creation platform for showcasing communication artifacts

Hands-On AI Activity

Students create a personal E-portfolio (using Wix or Adobe Portfolio) documenting 5 communication artifacts from all 5 units: (1) corrected grammar exercise (Unit I), (2) paraphrased essay with Hemingway score (Unit II), (3) professional email and resume (Unit III), (4) presentation slides and Orai report (Unit IV), (5) cultural communication reflection essay (Unit V). Each artifact must include an AI tool feedback snapshot. The portfolio is presented in the final class, peer-reviewed using a structured rubric, and checked for plagiarism via Copyleaks. Reflection summary (300 words) on AI tool usage across the course is mandatory.

Overall AI Tools Used in the Course

The table below consolidates all AI tools integrated across units, with direct URLs and their primary pedagogical applications.

#	AI Tool	URL / Platform	Primary Application in Course
1	ChatGPT (OpenAI)	https://chat.openai.com	Grammar, email drafting, essay structuring, content summarization
2	Grammarly	https://www.grammarly.com	Real-time grammar, spelling, tone detection, and writing suggestions
3	QuillBot	https://quillbot.com	Paraphrasing, summarization, and sentence restructuring
4	Hemingway Editor	https://hemingwayapp.com	Readability scoring and writing clarity improvement
5	Gamma.app	https://gamma.app	AI-powered presentation, document, and webpage creation
6	Orai (Speech Coach)	https://www.orai.com	AI speech coaching – fluency, pacing, filler-word analysis
7	Otter.ai	https://otter.ai	Auto-transcription and analysis of spoken communication
8	Canva AI	https://www.canva.com	Visual communication, poster, infographic, and slide design
9	Kickresume / Resume.io	https://www.kickresume.com	AI-assisted resume and cover letter generation

#	AI Tool	URL / Platform	Primary Application in Course
10	Copyleaks / Turnitin	https://copyleaks.com	Plagiarism detection and academic integrity checking
11	DeepL Translator	https://www.deepl.com	High-accuracy translation for intercultural communication
12	Beautiful.ai	https://www.beautiful.ai	AI-driven professional slide deck design and layout
13	Google Speech-to-Text	https://cloud.google.com/speech-to-text	Voice recognition for pronunciation and phonetics practice
14	Notion AI	https://www.notion.so/product/ai	Note-taking, report structuring, and knowledge organization

6. Prescribed Textbooks

S.No	Author(s)	Title	Publisher	Edition/Year
TB1	Meenakshi Raman & Sangeetha Sharma	Technical Communication: Principles and Practice	Oxford University Press	4th Edition, 2022
TB2	Jyoti Shetty & Rajesh Pareek	Communicative English for Engineers and Professionals	Wiley India	3rd Edition, 2021

7. Reference Books

S.No	Author(s)	Title	Publisher	Edition
R1	Rizvi, M. Ashraf	Effective Technical Communication	Tata McGraw-Hill	2nd Ed., 2018
R2	Murphy, Herta A.	Effective Business Communication	McGraw-Hill Education	7th Ed., 2019
R3	Guffey, Mary Ellen & Dana Loewy	Essentials of Business Communication	Cengage Learning	11th Ed., 2021
R4	Sethi, Anjanee& Bhavana Adhikari	Business Communication	Tata McGraw-Hill	3rd Ed., 2020
R5	Wren, P.C. & Martin, H.	High School English Grammar and Composition	S. Chand	Revised Ed., 2019
R6	Tickoo, M.L. & Suresh Kumar, E.R.	Writing with a Purpose	Oxford University Press	6th Ed., 2020

8. Online Learning Resources

#	Resource Name	URL	Description
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/109104101	IIT/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

Recommended YouTube Resources

#	Video Title	YouTube URL	Topic Area
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

11. CO-PO Mapping (Bloom's Taxonomy-Based)

Mapping Scale: 3 = Strong / High Correlation | 2 = Moderate Correlation | 1 = Weak / Indirect Correlation | – = No Direct Correlation

CO	Bloom's Level	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	Remember	1	-	-	1	-	1	-	1	1	3	1	2
CO2	Understand	1	-	-	1	1	2	1	1	2	3	1	2
CO3	Apply	-	-	1	-	3	2	1	2	3	3	2	2
CO4	Analyze	-	1	1	2	2	2	1	2	2	3	2	2
CO5	Evaluate	-	1	1	2	3	2	2	3	2	3	2	3
CO6	Create	-	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

Colour Key: Dark Green = 3 (High) Blue = 2 (Medium) Gold = 1 (Low) White = Not Mapped

Course Objectives:

- Understand quantum chemistry principles relevant to computing systems.
- Study semiconductor chemistry and fabrication processes.
- Analyze electrochemical energy storage systems for computing devices.
- Explore corrosion mechanisms in electronic systems and protection methods.
- Apply AI/ML tools for materials analysis, semiconductor modeling, and nanotechnology applications.

Course Code	APPLIED CHEMISTRY FOR COMPUTER SCIENCE	L	T	Pr	P	Credits
26AHS00101T	AI & DS | I Year II Sem	3	0	0	0	3
Prerequisite:	Basic Chemistry (Higher Secondary level)					

Prerequisites: Basic Chemistry (Higher Secondary level)

Course Objectives

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

Course Outcomes (COs)

After successful 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 applications.	**L3–L4**
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**

Course Content

UNIT – 1	Quantum Chemistry	9 Hrs
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 – 2	**Semiconductor Chemistry**	**9 Hrs**
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 – 3	**Electrochemistry and Energy Storage**	**9 Hrs**
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 $LiCoO_2$. • Supercapacitors: definition, classification, and applications. • Fuel cells: construction, working, and applications of hydrogen fuel cells 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 – 4	**Chemistry of Corrosion and Protective Techniques**	**9 Hrs**
Content	• Corrosion: definition, examples, types — chemical corrosion and electrochemical corrosion; Pilling-Bedworth rule. • Galvanic corrosion; concentration cell corrosion; factors affecting corrosion. • 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 – 5	**Chemistry of Nanomaterials**	**9 Hrs**
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.	

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L3	Apply	**CO1, CO2**	Molecular orbital energy diagram construction; semiconductor band gap and doping calculations.
L4	Analyze	**CO3, CO4**	Battery electrochemical reaction analysis; corrosion mechanism case studies for PCB failure modes.
L5/L6	Evaluate/Create	**CO5**	Nanomaterial design for a given computing application; AI-assisted property prediction project.

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	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

CO-PO Correlation Highlights	CO1 and CO2 align strongly with PO1 (Engineering Knowledge) and PO2 (Problem Analysis) through quantum chemistry and semiconductor analysis, and to PO12 through emerging quantum and computing materials domains. CO3 maps to PO2, PO3, and PO5 through energy storage design and AI-based battery optimization, with PO7 reflecting the sustainability relevance of green energy technologies. CO4 aligns with PO2, PO4, and PO5 through corrosion analysis and AI-based failure prediction. CO5 shows the broadest correlation, mapping to PO3 (nanomaterial design), PO5 (AI tools), PO7 (sustainable electronics), PO8 (responsible nanomaterial use), and PO10 (design project reporting), with strong PSO1 and PSO2 alignment across all COs through direct relevance to AI and computing hardware.

LEARNING RESOURCES

Textbooks

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.

Reference 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.

Online Resources

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

I Year II Sem

Course Code	DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS	L	T	Pr	P	Credits
26AHS00203T	AI & DS | I Year II Sem	3	0	2	0	4
Prerequisite:	Linear Algebra and Calculus					

Prerequisites: Linear Algebra and Calculus

Course Objectives

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

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 (Analyze)**
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 (Analyze)**
CO5	Apply Green's, Stokes', and Divergence theorems to compute work, circulation, and flux and evaluate engineering field behaviour.	**L5 (Evaluate)**

Course Content

UNIT – 1	First-Order Ordinary Differential Equations	9 Hrs
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 and natural growth/decay models; electrical circuit models (RC and RL circuits); engineering applications of first-order ODEs.	

AI Integration	Python SymPydsolve 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. Free/Open-Source AI Tools: Python (SymPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; GNU Octave.

UNIT – 2	**Higher-Order Linear Differential Equations**	**9 Hrs**
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.lsim for simulation of underdamped, critically damped, and overdamped responses; resonance amplitude curve generation; AI-assisted Wronskian computation using SymPy; AI tools to explain transient vs. steady-state response. Free/Open-Source AI Tools: Python (NumPy, SciPy, SymPy, Matplotlib); GNU Octave; WolframAlpha free tier.	

UNIT – 3	**Partial Differential Equations**	**9 Hrs**
Content	• Introduction and classification of PDEs; formation by elimination of arbitrary constants; formation by elimination of arbitrary functions; first-order linear PDEs and Lagrange's method (auxiliary equations); homogeneous linear PDEs with constant coefficients (CF and PI). • Standard forms of nonlinear first-order PDEs; second-order PDE applications — 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. Free/Open-Source AI Tools: Python (SymPy, NumPy, SciPy, Matplotlib); WolframAlpha free tier; Scilab.	

UNIT – 4	**Vector Differentiation**	**9 Hrs**
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 of gradient, divergence, and curl 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; AI tools to interpret Maxwell's equations in terms of divergence and curl operators. Free/Open-Source AI Tools: Python (NumPy, SymPy, Matplotlib, mpl_toolkits); SciPy for numerical computation.	

UNIT – 5	Vector Integration and Integral Theorems	9 Hrs
Content	• Line integrals over scalar and vector fields; work done and path independence; surface integrals and flux; Green's theorem in the plane and its applications. • Stokes' theorem; Divergence theorem (Gauss's theorem); applications to area and volume computation; engineering applications in electromagnetism, fluid mechanics, and mass conservation; relationship between integral theorems and Maxwell's equations.	
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. Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib); SymPy for symbolic integral verification; WolframAlpha free tier.	

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, CO4	Higher-order ODE response analysis for LCR circuits; gradient/divergence/curl field interpretation.
L5	Evaluate	CO5	Green's/Stokes'/Divergence theorem application for flux and circulation computation.

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	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.

LEARNING RESOURCES

Textbooks

1. Kreyszig, E. (2018). Advanced Engineering Mathematics (10th ed.). Wiley.
2. Grewal, B. S. (2017). Higher Engineering Mathematics (44th ed.). Khanna Publishers.

Reference Books

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

Online Resources

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

Web Resources & NPTEL Links

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

Recommended Free & Open-Source AI/Computational Tools

- Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for ODE/PDE solving, vector field visualization, and numerical integration.
- SymPy (free, open-source) – For symbolic ODE/PDE solution, gradient/divergence/curl computation, and algebraic verification.
- Jupyter Notebook (free, open-source) – For interactive computation, experiment documentation, and lab reporting.
- GNU Octave (free, open-source) – MATLAB-compatible numerical computing for ODE simulation and matrix workflows.
- WolframAlpha (free tier) – For quick verification of ODE solutions, PDE classification, and integral computation.

- Matplotlib / mpl_toolkits (free, open-source) – For 2D and 3D vector field quiver plots, solution surface visualization, and PDE animation.

- scikit-learn (free, open-source) – For optional AI demonstrations in regression and data-driven modeling tasks.

I Year –II Semester

Course Code	DATA STRUCTURES & EFFICIENT PROBLEM SOLVING	L	T	Pr	P	Credits
26ACS05201T	Common to CSE and Allied Branches	3	0	0	0	3

Course Objectives

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

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)

Course Content

UNIT – I	INTRODUCTION, ALGORITHM ANALYSIS, AND LINEAR STRUCTURES	9 Hrs

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	9 Hrs

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	9 Hrs

Content	□ Singly Linked List: node structure, insertion at head/tail/position, deletion by value/position, 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	9 Hrs

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	9 Hrs

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.

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 PrjMgm
CO1	L3	3	3	1	2	2	–	–	–	–	1	–
CO2	L3	3	3	2	2	2	–	–	–	–	1	–
CO3	L4	3	3	2	2	2	–	–	–	1	1	–
CO4	L4	3	3	3	2	2	–	–	–	1	1	–
CO5	L4	3	3	3	3	2	–	–	–	1	2	–

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.

LEARNING RESOURCES

Textbooks

1. TB1: Shriram K. Vasudevan, Abhishek S. Nagarajan, Karthick Nanmaran, Data Structures Using Python, Wiley India.
2. TB2: Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Data Structures and Algorithms in Python (Indian Adaptation), Wiley India.

Reference Books

1. Brad Miller and 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
- VisuAlgo – visualgo.net | Python Tutor – pythontutor.com |NetworkX – networkx.org
- GeeksforGeeks – geeksforgeeks.org | Algorithm Visualizer – algorithm-visualizer.org
-

1. Course Code	ENGINEERING GRAPHICS	L	T	Pr	P	Credits
26AME03201T	AI & DS | I Year I Sem	1	0	4	0	3
Prerequisite:	Basic Geometry, Fundamental Python Programming					

Prerequisites: Basic Geometry, Fundamental Python Programming

Lab Course Outcomes

After successful completion of the lab, 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.	**L4 (Analyze)**
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.	**L4 (Analyze)**
CO5	Produce composite isometric layouts alongside generative text-to-CAD system alternatives for dynamic prototyping.	**L5 (Evaluate)**

List of Experiments

Unit 1	Introduction to Engineering Graphics, Curves and Scales

1. Manual Construction of Engineering Curves

Construct an ellipse using the eccentricity method and general method given the major and minor axes; construct a parabola and a hyperbola using the rectangle/eccentricity method. Verify dimensional accuracy through measurement and document construction steps.

2. Cycloidal Curves and Involutes Construction

Draw a cycloid generated by a rolling circle and an involute of a given circle/polygon using standard geometric construction methods. Annotate the generating points and verify the curve trace against theoretical points.

3. Construction of Plain, Diagonal and Vernier Scales

Construct a plain scale, a diagonal scale, and a vernier scale for given representative fractions and measure specified distances using each. Cross-verify readings between manually constructed scales and a digital scale tool.

4. AI-Assisted Conic Curve Generation and CV Edge Detection

Use AI prompt engineering to generate code blocks that algorithmically plot conic curves (ellipse, parabola, hyperbola) and compare with the manually drafted curve. Capture a hand-drawn geometric sketch and apply Computer Vision edge detection (Canny Edge/Hough Transform) to extract a clean vector shape.

Unit 2	Orthographic Projections of Points, Lines and Planes

1. Projection of Points in All Four Quadrants

Draw the orthographic projections (top view and front view) of points located in each of the four quadrants relative to the horizontal and vertical reference planes. Verify the position of each projection against the given point coordinates.

2. Projection of Lines Inclined to Reference Planes

Determine the projections of a straight line inclined to one or both reference planes; find the true length and true inclination of the line using the rotation or auxiliary plane method. Cross-check results using trigonometric calculation.

3. Projection of Planes in Various Orientations

Draw projections of regular planes (triangle, square, pentagon, hexagon) positioned perpendicular to both reference planes, parallel to one reference plane, and inclined to both reference planes. Verify true shape recovery using the auxiliary view method.

4. 3D Coordinate Estimation Script from 2D Projections

Develop a transformation script where a user inputs 2D projection endpoints (top view and front view coordinates) and the script automatically computes the 3D line representation, true spatial length, and inclination angles. Validate the script output against a manually solved projection problem.

Unit 3	Projection of Regular Solids

1. Projection of Solids with Axis Perpendicular to a Reference Plane

Draw the projections of a hexagonal prism and a cone with the axis perpendicular to the horizontal plane, and repeat with the axis perpendicular to the vertical plane. Verify true shape and hidden line conventions in each case.

2. Projection of Solids with Axis Inclined to a Reference Plane

Draw the projections of a square pyramid with its axis inclined to one reference plane and parallel to the other, using the change-of-position or auxiliary plane method. Verify the apparent and true lengths of the solid's edges.

3. Solid Classification Using a Machine Learning Notebook

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 a given orthographic engineering layout image. Test the classifier against five different solid images and document accuracy.

Unit 4	Sections of Solids and Development of Surfaces

1. Sectional Views of Solids with Inclined Cutting Planes

Draw the sectional view and true shape of section for a pyramid and a cone cut by a section plane inclined to the base, for solids in simple position. Verify the true shape of the cut surface using the auxiliary projection method.

2. Development of Lateral Surfaces — Parallel Line Method

Develop the lateral surface of a cylinder and a prism using the parallel-line development method. Verify the unfolded pattern dimensions against the original solid's surface measurements.

3. Development of Lateral Surfaces — Radial Line Method

Develop the lateral surface of a cone and a pyramid using the radial-line development method. Verify the unfolded pattern by reconstructing a physical or digital model from the developed surface.

4. Sheet Nesting Optimization Using Heuristic Algorithms

Use a nesting optimization script (genetic algorithm or standard heuristic) to arrange multiple custom flat-surface solid developments onto a finite raw sheet grid, minimizing material margins and cut lines. Compare the optimized layout with a manually arranged layout in terms of material utilization.

Unit 5	Isometric Projections and Computer Graphics

1. Isometric Projection of Simple and Compound Solids

Construct the isometric view of a simple solid (cube, prism, pyramid) using the isometric scale, and extend to a compound solid formed by combining two basic solids. Verify isometric axis angles and dimensional accuracy.

2. Conversion Between Orthographic and Isometric Views

Given a set of orthographic views (front, top, side) of an object, construct the corresponding isometric view; conversely, given an isometric view, derive the orthographic views. Cross-verify both conversions for dimensional consistency.

3. Computer-Aided Drafting Using AutoCAD

Create 2D drawings of simple machine components and a basic PCB layout using AutoCAD; apply transformation commands (move, rotate, scale, mirror) to manipulate the drawn entities. Export and document the final drawings (not for end examination).

4. 2D-to-3D Depth Reconstruction Using a Pretrained Model

Use a pretrained depth-estimation or pixel-to-voxel neural model to parse a set of front, top, and side drawings, compiling them into an interactive 3D point cloud or boundary surface rendering. Compare the AI-reconstructed 3D model with the manually constructed isometric drawing.

AI Tools Integration (Lab)

AI Tools Integration (Lab)	AI prompt engineering for generating geometric profiles and algorithmic conic curve code; Computer Vision (OpenCV — Canny Edge/Hough Transform) for parsing hand-drawn sketches into vector shapes. Machine learning pipelines (scikit-learn, simple CNNs) for 3D coordinate estimation and solid-type classification from orthographic images. Generative Adversarial Networks (GANs) and heuristic/genetic algorithms for flat-pattern nesting optimization. Neural Radiance Fields (NeRF) and pixel-to-voxel models for 2D-to-3D depth reconstruction from multi-view engineering sketches. AutoCAD for computer-aided 2D/3D drafting and transformations.

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	–	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

CO-PO Correlation Highlights	CO1 and CO2 strongly align with PO1 (Engineering Knowledge) and PO2 (Problem Analysis) through traditional drafting validated with algorithmic verification. CO3 and CO4 connect to PO3 (Design) and PO5 (Modern Tools) through ML-based solid classification and nesting optimization. CO5 demonstrates the strongest PO5 alignment, integrating generative AI text-to-CAD concepts with traditional isometric drafting skills.

LEARNING RESOURCES

Textbooks

1. Bhatt, N. D. (2016). Engineering Drawing. Charotar Publishing House.
2. Jolhe, D. (2017). Engineering Drawing with an Introduction to AutoCAD. Tata McGraw Hill.

Reference Books

1. Jain, P., Gautam, A. P., & Bhargava, A. (n.d.). Engineering Graphics and Design. Khanna Publishing.
2. Narayana, K. L., &Kannaiah, P. (2013). Engineering Drawing. Tata McGraw Hill.
3. Shah, M. B., & Rana, B. C. (2009). Engineering Drawing. Pearson Education.

Online Resources

- NPTEL Engineering Graphical Design Course – IIT Kharagpur
- OpenCV Official Python Bootcamps and Tutorials

MIT 6.S192: Deep Generative Modeling Platforms

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 Platforms

I Year I Sem

• Course Code	ENGINEERING WORKSHOP	L	T	Pr	P	Credits
26AME03201P	AI & DS | I Year I Sem	0	0	0	3	1.5
Prerequisite:	Basic Computing Elements					

Prerequisites: Basic Computing Elements

Lab Course Outcomes

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recognize the importance of safety precautions while working in the workshop.	**L2 (Understand)**
CO2	Identify different measuring, marking, cutting tools used in the workshop and their operational capabilities.	**L2–L3**
CO3	Fabricate precision wood and metal fitting joints within standard specifications.	**L3 (Apply)**
CO4	Understand wiring standards, safe residential switches, lighting circuits, and basic appliances.	**L2–L3**
CO5	Identify different tools used in foundry and welding sections.	**L2 (Understand)**

List of Experiments

Unit 1	Safety Practices, Carpentry and Fitting Trades

1. Workshop Safety Orientation and PPE Demonstration

Demonstrate proper use of Personal Protective Equipment (helmet, safety glasses, gloves, apron) and identify common workshop hazards. Practise emergency protocols including fire extinguisher use and first-aid station location, using computer vision-based safety compliance monitoring as a demonstration.

2. Carpentry — Half-Lap and Mortise & Tenon Joints

Prepare a half-lap joint and a mortise-and-tenon joint using basic carpentry tools (saw, chisel, mallet, marking gauge) on a wooden workpiece. Verify joint fit and dimensional accuracy against the given drawing specification.

3. Carpentry — Corner Dovetail or Bridle Joint

Prepare a corner dovetail joint or bridle joint following standard carpentry marking and cutting procedures. Inspect the finished joint for fit, squareness, and surface finish.

4. Fitting — V-Fit and Dovetail Fit

Perform a V-fit and a dovetail fit exercise on mild steel workpieces using fitting tools (files, scriber, try square, vernier caliper). Verify dimensional accuracy and fit tolerance against the drawing.

5. Fitting — Semi-Circular Fit

Perform a semi-circular fit exercise combining male and female workpieces to standard tolerance. Inspect for fit accuracy using feeler gauges and visual inspection.

Unit 2	Sheet Metal, Plumbing and Welding Trades

1. Sheet Metal Working — Tapered Tray and Conical Funnel

Develop and fabricate a tapered tray and a conical funnel from GI sheet using sheet metal tools (snips, mallet, stakes); apply development of surfaces principles for accurate pattern marking. Verify dimensional accuracy of the finished components.

2. Sheet Metal Working — Elbow Pipe and Brazing

Develop and fabricate an elbow pipe joint from GI sheet, and perform a brazing exercise to join two metal pieces. Inspect the joint strength and surface finish of the brazed connection.

3. Electrical Wiring — Series, Parallel and Two-Way Switch Circuits

Make electrical connections for a parallel and series lighting circuit, and wire a two-way switch circuit for staircase lighting control. Test each circuit for correct operation and safety compliance using a multimeter.

4. Electrical Wiring — Godown Lighting, Tube Light and Soldering

Wire a godown lighting circuit and a tube light circuit (including choke and starter connections); practise soldering of wire joints. Test all connections for continuity and correct functioning.

5. Plumbing — Pipe Joint Preparation

Prepare a simple PVC pipe joint with coupling for same-diameter pipes and with a reducer for different-diameter pipes, using thread cutting and pipe-fitting tools. Test the joint for leak-tightness using a pressure or water test.

6. Welding Shop — Lap Joint and Butt Joint

Demonstrate and practise Arc Welding and Gas Welding techniques to prepare a simple lap joint and a butt joint on mild steel plates. Inspect the welded joint for uniformity, penetration, and visual defects.

Unit 3	Foundry, Computer Hardware and Smart Systems

1. Foundry Trade — Green Sand Mould Preparation

Demonstrate and practise moulding tools and processes to prepare a green sand mould for a simple pattern. Inspect the mould cavity for surface finish and dimensional accuracy before pouring simulation.

2. Introduction to 3D Printing

Demonstrate the core concepts of 3D printing including hardware components, software workflow, digital file preparation (STL/G-code), and printing materials. Observe the 3D printing process for a simple complex-geometry object and document the print quality.

3. Computer Hardware Disassembly and Reassembly

Disassemble and reassemble a desktop computer to understand the role of each component (CPU, motherboard, RAM, HDD/SSD, GPU). Document the function and interconnection of each component and verify successful reassembly through a boot test.

4. Hardware Troubleshooting and Peripheral Installation

Simulate common hardware issues (RAM failure, CPU overheating) and identify the procedures for resolving each. Install and configure peripheral devices such as a printer, scanner, or external hard drive, and verify successful operation.

5. Network Setup and Configuration

Set up a small LAN network with routers, switches, and computers; configure IP addresses, subnet masks, and basic network services. Verify network connectivity between devices using ping tests.

6. Disassembly and Assembly of Mechanical Equipment

Familiarise with the disassembly and assembly process of a simple mechanical system such as a bicycle gear box or braking system. Document the function of each component and verify correct reassembly through operational testing.

7. Smart Energy Monitoring Using IoT Sensors

Set up an IoT current sensor to monitor energy consumption of a simple electrical circuit; log the data and build a basic machine learning load forecasting model. Train an anomaly detection routine to flag unusual electrical load patterns (simulated arc/leakage events).

AI Tools Integration (Lab)

AI Tools Integration (Lab)	Computer vision for safety compliance monitoring in the workshop environment. Smart energy monitoring using IoT current sensors with machine learning load forecasting models. Training of anomaly detection systems to automatically flag electrical arc or leakage patterns, supporting predictive load-monitoring model development. Python-based data logging and visualization for sensor data collected during the IoT energy monitoring experiment.

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	–	–	–	–	–	–	2
CO2	3	2	2	1	2	–	–	–	–	–	–	1
CO3	3	1	–	–	1	2	–	–	–	–	–	2
CO4	–	–	–	–	–	–	–	1	3	–	1	–
CO5	–	–	–	–	–	–	–	–	–	3	–	1

CO-PO Correlation Highlights	CO1 maps strongly to PO1 and PO5 through safety awareness reinforced by computer vision compliance demonstrations. CO2 and CO3 connect to PO1, PO2, PO3, and PO5 through hands-on fabrication exercises. CO4 links to PO8 (Ethics) and PO9 (Teamwork) through electrical safety practice. CO5 maps to PO10 (Communication) and PO11 (Project Management) through foundry and computer hardware documentation tasks.

LEARNING RESOURCES

Textbooks

1. Hajra Choudhury, S. K. (n.d.). Workshop Technology Vol I & II. Media Promoters & Publishers.
2. Raghuwanshi, B. S. (2015/2017). A Course in Workshop Technology. Dhanpat Rai & Co.

Reference Books

1. Veneri, G., & Capasso, A. (n.d.). Hands-On Industrial Internet of Things. Packt Publishing.
2. Felix, W. (2019). Basic Workshop Technology: Manufacturing Process. Independently Published.
3. Black, B. J. (2015). Workshop Processes, Practices and Materials (5th ed.). Routledge.
4. Rao, P. N. (n.d.). Manufacturing Technology (Foundry, Forming and Welding). McGraw Hill.
5. Mobley, R. K. (n.d.). An Introduction to Predictive Maintenance. Butterworth-Heinemann.

Online Resources

- NPTEL Workshop Practices Video Series – IIT Kharagpur
- Predictive Maintenance Models in Python Tutorial

MIT 6.131: Power Electronics and Smart Grid Interfaces

Course Code	APPLIED CHEMISTRY FOR COMPUTER SCIENCE LAB	L	T	Pr	P	Credits
26AHS00101P	AI & DS | I Year II Sem	0	0	0	3	1.5
Prerequisite:	Applied Chemistry for Computer Science (co-requisite)					

Prerequisites: Applied Chemistry for Computer Science (co-requisite)

Lab Course Outcomes

After successful completion of the lab, 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**

List of Experiments

Unit 1	Titrimetric and Conductometric Analysis

1. Estimation of Ferrous Iron (Fe2+) using Potassium Permanganate (KMnO4) — Redox Titration
Standardize a KMnO4 solution against a known oxalate solution, then titrate a Fe2+ sample solution against the standardized KMnO4 until a persistent pink endpoint is observed. Calculate the concentration and amount of Fe2+ present in the given sample using stoichiometric relations.

2. Determination of Strength of Acid (H2SO4) in Battery Using Sodium Hydroxide (NaOH)
Standardize a NaOH solution against a primary standard, then titrate a diluted battery acid sample against the standardized NaOH using a suitable indicator. Determine the strength and concentration of sulfuric acid present in the battery sample.

3. Determination of Corrosion
Expose metal coupons (mild steel or copper) to a corrosive medium for a fixed duration, measure weight loss before and after exposure, and calculate the corrosion rate using the weight-loss method. Document observations on surface changes.

4. Determination of Corrosion by Environment
Expose identical metal coupons to different environmental conditions (humid, saline, acidic) and compare corrosion rates across environments. Analyse which environmental factor accelerates corrosion most significantly and relate findings to real-world electronic hardware protection needs.

Unit 2	Conductometric and Potentiometric Titrations

1. Conductometric Titration of Strong Acid vs Strong Base (HCl vs NaOH)

Titrate HCl against NaOH while measuring conductivity at regular intervals using a conductivity meter. Plot conductance versus volume of titrant added, identify the equivalence point from the graph, and determine the concentration of the acid.

2. Conductometric Titration of Weak Acid vs Strong Base (CH3COOH vs NaOH)

Titrate acetic acid against NaOH while recording conductivity readings. Plot the conductometric titration curve, identify the characteristic curve shape for a weak acid–strong base titration, and determine the equivalence point and acid concentration.

3. Potentiometry (Redox Titration) — Estimation of Iron (Fe2+) Using Potassium Dichromate (K2Cr2O7)

Set up a potentiometric titration using a platinum electrode and reference electrode, titrate a Fe2+ solution against standardized K2Cr2O7, and record EMF values at regular intervals. Plot EMF versus volume to identify the equivalence point and calculate Fe2+ concentration.

Unit 3	Spectroscopic Analysis and Polymer Synthesis

1. Spectrophotometry — Determination of Wavelength (λ) and Verification of Beer-Lambert's Law

Prepare a series of standard solutions of a coloured compound at varying concentrations, measure absorbance at a fixed wavelength using a UV-Vis spectrophotometer, and plot absorbance versus concentration to verify Beer-Lambert's law. Determine the molar extinction coefficient from the slope.

2. Identification of Functional Groups in Organic Compound by Infra-Red (IR) Spectrum

Record the IR spectrum of a given organic compound and identify characteristic absorption bands corresponding to functional groups (C=O, O-H, N-H, C-H, etc.). Compare the spectrum with reference IR charts to confirm the identity of functional groups present.

3. Preparation of Polymer (Phenol-Formaldehyde — Bakelite)

Synthesize Bakelite by the controlled condensation reaction of phenol and formaldehyde under acidic or basic catalytic conditions, observing the resin formation stages (A-stage, B-stage). Document the reaction conditions and resulting polymer properties, and analyse the reaction's industrial relevance to electronic component insulation.

AI Tools Integration (Lab)

AI Tools Integration (Lab)	Data analysis using Python (NumPy, Pandas, Matplotlib) for titration curve plotting and equivalence point detection. Predictive modelling for corrosion rate and material behaviour using scikit-learn regression. TensorFlow for optional IR spectrum pattern classification. OpenCV for colorimetric image-based concentration estimation. Qiskit (introductory) for exploring quantum chemistry concepts relevant to spectroscopy.

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	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
Average	2.0	-	-	2.2	3.0	2.0	1.6	-	-	-	-	-	2.4	2.6

CO-PO Correlation Highlights	CO1 and CO3 map strongly to PO1 and PO2 through titration-based quantitative analysis. CO2 connects to PO4 (Investigation) through corrosion rate determination under varying environmental conditions. CO5 links to PO5 (Modern Tools) through Python-based predictive modelling of material behaviour and polymer synthesis documentation.

LEARNING RESOURCES

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

ASME technical resources

YOUTUBE CHANNELS

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

NPTEL / SWAYAM COURSES

Course Name	Platform	Provider	Link

Developing Soft Skills and Personality	NPTEL / SWAYAM	IIT Kanpur	https://swayam.gov.in
Business English Communication Suite	Coursera (Free Audit)	University of Washington	https://coursera.org
Speak English Professionally: In Person, Online & On the Phone	Coursera	Georgia Tech	https://coursera.org
English and Academic Preparation	NPTEL	IIT Bombay	https://nptel.ac.in
Technical English for Engineers	NPTEL / SWAYAM	IIT Roorkee	https://swayam.gov.in

LAB SOFTWARE & AI TOOLS REQUIRED

Software / Tool	Type	Purpose
ELSA Speak	AI Pronunciation Coach App	Real-time speech accuracy & fluency feedback
Grammarly	AI Writing Assistant	Grammar, punctuation, style correction
ChatGPT / Claude.ai	Generative AI Chatbot	Conversation practice, writing assistance, debate prep
Otter.ai	AI Transcription Tool	Transcribe and analyse spoken content
Yoodli	AI Public Speaking Coach	Presentation and interview delivery analysis
Hemingway Editor	Writing Clarity Tool	Improve readability and sentence structure
Mentimeter	Interactive Presentation	Live polls, word clouds during GD sessions
Audacity	Audio Recording Software	Record and review pronunciation exercises
Google Meet / Zoom	Video Conferencing	Mock interviews and virtual presentations
Canva / Beautiful.ai	AI Presentation Design	Create professional visual presentations

I Year –II Semester

Course Code	DATA STRUCTURES & EFFICIENT PROBLEM SOLVING LAB	L	T	Pr	P	Credits
26ACS05201P	Common to CSE and Allied Branches	0	0	0	2	1

Prerequisites: Programming fundamentals in Python. (Inferred – not specified in the source.)
Branch: Common to CSE and Allied Branches | Regulation: JNTUA R26

Course Objectives

- Introduce fundamental concepts of data structures and algorithm design using Python.
- Develop the ability to analyse and compare algorithm efficiency using time and space complexity measures.
- Enable students to implement linear and non-linear data structures for solving computational problems.
- Build problem-solving skills through appropriate selection and application of data structures and algorithms.
- 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:

- **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.
- **Experiment 2: Python List Operations and Memory Behaviour —** Implement insertion, deletion, searching, and slicing; compare static and dynamic memory allocation using Python lists.
- **Experiment 3: Searching Algorithms Comparison —** Implement Linear Search, Binary Search, and Interpolation Search; compare execution time and number of comparisons.
- **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:

- **Experiment 5: Stack Implementation and Applications —** Implement stack operations using Python lists and solve parenthesis-matching problems.
- **Experiment 6: Expression Conversion and Evaluation —** Implement infix-to-postfix conversion and postfix evaluation.
- **Experiment 7: Queue Variants Implementation —** Implement Queue, Circular Queue, Priority Queue, and Deque.
- **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:

- **Experiment 9: Singly Linked List Operations —** Implement insertion, deletion, traversal, and reversal operations.
- **Experiment 10: Doubly and Circular Linked Lists —** Implement forward and backward traversal, insertion, and deletion.
- **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:

- **Experiment 12: Binary Tree Construction and Traversals —** Implement inorder, preorder, postorder, and level-order traversals.
- **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:

- **Experiment 14: Graph Representation and Traversal —** Represent graphs using adjacency matrix and adjacency list; implement BFS and DFS.
- **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 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	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 & 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 & Associates.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, Pearson Education.
3. Narasimha Karumanchi, *Data Structures and Algorithms Made Easy*, CareerMonk Publications.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein, *Introduction to Algorithms*, MIT Press.
5. Aditya Bhargava, *Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People*, Manning Publications.

Online Learning Resources

- NPTEL – Data Structures and Algorithms – nptel.ac.in
- VisuAlgo – visualgo.net
- Python Tutor – pythontutor.com
- Algorithm Visualizer – algorithm-visualizer.org
- NetworkX Documentation – networkx.org
- GeeksforGeeks – Data Structures – geeksforgeeks.org
- 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.

I Year II Sem

Course Code	HEALTH & WELLNESS, YOGA AND SPORTS	L	T	Pr	P	Credits
26AHS00201L	Common to All Branches	0	0	0	1	0.5

Course Objectives

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

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)

Unit-wise Syllabus with Learning Outcomes

UNIT 1	Health, Wellness and Preventive Healthcare	5 Hrs
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	**On completion of this unit, students will be able to:** • ULO1.1: Define health, wellness and fitness and describe the seven dimensions of	

Outcomes (ULOs)	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	5 Hrs
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	5 Hrs
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]

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

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

YouTube Resources

- Surya Namaskar – Complete 12-Step Tutorial – https://youtu.be/surya_tut_01
- Pranayama for Beginners (Swami Ramdev) – https://youtu.be/pranayama_sr_02
- Mindfulness Meditation (10 Min) – https://youtu.be/mindful_med_03
- Balanced Nutrition Explained (ICMR) – https://youtu.be/nutrition_icmr_04
- Sports Injury Prevention Tips – https://youtu.be/injury_prev_05

- Mental Health Awareness for Students – https://youtu.be/mh_students_06
- Yoga for Stress Relief – Isha Foundation – https://youtu.be/yoga_isha_07
- 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.

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	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	-

Course Code	PROBABILITY AND STATISTICS	L	T	Pr	P	Credits
26AHS00301T	AI & DS | II Year I Sem	3	0	0	3	3
Prerequisite:	Linear Algebra and Calculus; Differential Equations and Vector Calculus					

Prerequisites: Linear Algebra and Calculus; Differential Equations and Vector Calculus

Course Objectives

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

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 describxxxxxxxxxxxxxe 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)**

Course Content

UNIT – 1	Descriptive Statistics	9 Hrs
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 – 2	Probability and Random Variables	9 Hrs

Content	• Sample space and events; axioms of probability; addition and multiplication theorems; conditional probability; Bayes' theorem. • 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 – 3	**Probability Distributions**	**9 Hrs**
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: Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy; Scilab.	

UNIT – 4	**Sampling Theory and Estimation**	**9 Hrs**
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 – 5	**Testing of Hypothesis and Correlation**	**9 Hrs**
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); statsmodels library.	

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	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
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

LEARNING RESOURCES

Textbooks

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

Reference Books

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

Online Resources

- NPTEL: Probability and Statistics – https://nptel.ac.in
- MIT OpenCourseWare – Introduction to Probability and Statistics: https://ocw.mit.edu
- Python SciPy Statistics Documentation: https://docs.scipy.org/doc/scipy/reference/stats.html
- Statsmodels Documentation: https://www.statsmodels.org

1. Google Colab for Python: https://colab.research.google.com

Web Resources & NPTEL Links

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

Recommended Free & Open-Source AI/Computational Tools

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

II Year II Sem

Course Code	BIOLOGY FOR ENGINEERS	L	T	Pr	P	Credits
26AHS00304T	Common to All Branches	2	0	0	0	2

Course Outline

This course introduces biological concepts through the lens of engineering and artificial intelligence. It emphasizes biomolecules, human systems, bio-inspired designs, and modern bioengineering trends, with practical case studies and AI-enabled problem-solving.

Course Objectives (for teachers)

- To provide foundational biological knowledge with engineering relevance.
- To enable teachers to demonstrate AI-driven applications in biology and bioengineering.
- To encourage interdisciplinary teaching that connects biology, engineering, and computational tools.
- To prepare students for innovation in healthcare, materials, and bio-industries.

Course Content

Unit 1: Foundations of Biology and AI

— Core Content	— AI Integration
Cell as the basic unit of life. Carbohydrates, proteins, lipids, enzymes, vitamins, hormones.	AI-based visualization of biomolecules (e.g., AlphaFold). Case studies on AI in drug discovery.
Free/Open-Source AI Tools:	
Protein folding prediction using AlphaFold, Use of DeepChem to simulate enzyme-substrate binding, Compare AI-predicted protein structures with experimental data.	

Unit 2: Biomolecules in Engineering Applications

— Core Content	— AI Integration
Carbohydrates in bioplastics (PHA, PLA) Proteins in food engineering (whey protein, plant analogs)	AI-driven generative design for biomaterials. Lipids in biodiesel &detergents.
Free/Open-Source AI Tools:	
PLA bioplastic design using ANSYS, AI-based optimization of protein-rich diets, Design of a cellulose-based water filter using AI simulation, Model biodiesel yield prediction using MATLAB.	

Unit 3: Human Systems and Bio-Designs

— Core Content	— AI Integration
Brain as CPU (EEG, robotic prosthetics). Eye as camera (optical corrections, lens materials).	Heart as pump (ECG, stents). Kidney as filtration (dialysis systems).
Free/Open-Source AI Tools:	
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.	

Unit 4: Bio-Inspired Materials and Mechanisms

— Core Content	— AI Integration
Photosynthesis → photovoltaic cells. Bird flight → GPS and aircraft navigation.	Lotus leaf → superhydrophobic surfaces. Kingfisher beak → bullet train design.
Free/Open-Source AI Tools:	

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.

Unit 5: Emerging Trends in Bioengineering

— Core Content	— AI Integration
ꕥ Muscular and skeletal scaffolds. ꕥ 3D bioprinting of tissues (ear, bone, skin).	ꕥ AI in medical imaging (MRI, CT scans). ꕥ AI in diagnostics and predictive healthcare.
Free/Open-Source AI Tools:	
AI-based tumor detection in MRI scans, 3D bioprinting case study of bone tissue, Use of AI to classify medical images, To Design a scaffold for bone regeneration using ANSYS.	

Course Outcomes (for students)

- Apply biological concepts to engineering problems.
- Use AI tools to analyze biological data and design solutions.
- Evaluate biomimetic principles for real-world applications.
- Develop interdisciplinary approaches for socially relevant bioengineering challenges.

CO–PO Mapping for Biology for Engineers (AI Integrated)

Course Outcomes (COs)

CO Code	Course Outcome Statement
CO1	Explain the fundamentals of biology, biomolecules, and human systems relevant to engineering applications.
CO2	Apply AI tools and computational techniques for biological data analysis, biomolecule visualization, and healthcare applications.
CO3	Analyze biomolecules and bio-inspired systems for solving engineering problems and developing sustainable technologies.
CO4	Design biomimetic and bioengineering solutions using interdisciplinary engineering approaches and AI-enabled tools.
CO5	Evaluate emerging trends in bioengineering such as 3D bioprinting, regenerative medicine, and AI-based diagnostics for societal applications.

CO – PO Articulation Matrix

Correlation Levels:3– High; **2 – Moderate**; **1 – Low**, Blank – No correlation

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

Suggested Learning Resources:

Text Books

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

2. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., ThilagarajW.,Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.

Reference Books

- Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
- Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
- Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
- Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
- Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
- Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
- 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.

Web links and Video Lectures (e-Resources):

https://nptel.ac.in/courses/121106008
https://freevideolectures.com/course/4877/nptel-biology-engineers-other-non-biologists
https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009
https://ocw.mit.edu/courses/20-010j-introduction-to-bioengineering-be-010j-spring-2006
https://www.coursera.org/courses?query=biology
https://onlinecourses.nptel.ac.in/noc19_ge31/preview
https://www.classcentral.com/subject/biology
https://www.futurelearn.com/courses/biology-basic-concepts

AI Tools for the Course

- DeepChem – drug discovery simulations.
- AlphaFold – protein structure prediction.
- Generative Design AI – biomaterial innovation.
- ANSYS – biomimetic engineering simulations.
- MATLAB Simulink – biological system modeling.
- AI Medical Imaging Tools – diagnostics and healthcare.

Course Code	DIGITAL LOGIC AND COMPUTER ORGANIZATION	L	T	Pr	P	Credits
26AEC04301T	AI & DS | II Year I Sem	3	0	0	0	3
Prerequisite:	Computational Thinking and Problem Solving with Python					

Prerequisites: Computational Thinking and Problem Solving with Python

Course Objectives

- 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.
- Analyse and design combinational circuits and sequential circuits.
- Understand Von-Neumann architecture, functional units, and bus structures; apply computer arithmetic techniques.
- Understand memory organization covering RAM, ROM, cache mapping, virtual memory, and secondary storage; implement programmable logic using PLA, PAL, and FPGA.
- 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).

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**

Course Content

UNIT – 1	Data Representation and Digital Logic Circuits – I	9 Hrs
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 – 2	**Digital Logic Circuits – II and Basic Structure of Computers**	**9 Hrs**
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 – 3	**Computer Arithmetic**	**9 Hrs**
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 – 4	**Memory Organization and Programmable Logic**	**9 Hrs**
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 – 5	Input/Output Organization and Interfacing	9 Hrs
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.	

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.

LEARNING RESOURCES

Textbooks

1. Hamacher, C., Vranesic, Z., & Zaky, S. (2023). Computer Organization (6th ed.). McGraw Hill.
2. Mano, M. M. (2018). Digital Design (6th ed.). Pearson Education.

Reference Books

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

Online Resources

- NPTEL Computer Organization: https://nptel.ac.in
- NPTEL Digital Logic Design: https://nptel.ac.in
- Google Colab for Python: https://colab.research.google.com

VisuAlgo – Visualizing Data Structures: https://visualgo.net

Course Code	ALGORITHMS DESIGN AND ANALYSIS	L	T	Pr	P	Credits
26ACS05301T	Common to CSE & Allied Branches	3	0	0	0	3

Course Objectives

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

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)

Course Content

UNIT – I	ALGORITHM ANALYSIS AND MATHEMATICAL FOUNDATIONS	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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.

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 PrjMgm
CO1	L4	3	3	2	3	2	–	–	–	–	1	–
CO2	L4	3	3	2	3	2	–	–	–	–	1	–
CO3	L4	3	3	3	3	2	–	–	–	1	2	–
CO4	L5	3	3	3	3	3	–	–	–	1	2	1
CO5	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.

LEARNING RESOURCES

Textbooks

1. TB1: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
2. TB2: Jon Kleinberg and Eva Tardos, Algorithm Design, Pearson Education, 2013.

Reference Books

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, Algorithms, McGraw-Hill, 2008. (Free PDF at algorithms.wtf)
2. Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011.

Online Resources

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

Course Code	OBJECTORIENTED DESIGN AND PROGRAMMING	L	T	Pr	P	Credits
26ACS05302T	Common to CSE & Allied Branches	3	0	0	0	3

Course Objectives

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

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

Course Content

UNIT – I	OOP CONCEPTS, DATA TYPES, OPERATORS, AND CONTROL STATEMENTS	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

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.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define OOP principles; describe class, object, constructor, inheritance, interface.
L3	Apply	**CO3–CO5**	Implement inheritance hierarchies; apply exception handling; develop a JavaFX GUI.
L4	Analyse	**CO1, CO2, CO6**	Analyse static vs. dynamic dispatch; compare ArrayList vs. LinkedList.
L5	Evaluate	**CO2, CO6**	Evaluate collection choice for a given performance requirement.
L6	Create	**CO1, CO5**	Design and implement a full JDBC-backed JavaFX Student Management Application.

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	–
CO2	L4	3	3	3	2	2	–	–	–	1	1	–
CO3	L3	3	3	2	1	2	–	–	–	–	1	–
CO4	L3	3	3	2	1	2	–	–	–	–	1	–
CO5	L3	3	2	2	1	2	–	–	–	–	1	–
CO6	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.

LEARNING RESOURCES

Textbooks

1. TB1: Anitha Seth and B.L. Juneja, JAVA one step ahead, Oxford University Press.
2. TB2: Debasis Samanta and Monalisa Sarma, Joy with JAVA — Fundamentals of Object Oriented Programming, Cambridge University Press, 2023.
3. TB3: Paul Deitel and Harvey Deitel, Java 9 for Programmers, 4th Edition, Pearson Education.

Reference Books

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

Online Resources

- NPTEL – Programming in Java – nptel.ac.in/courses/106/105/106105191/
- Oracle Java Documentation – docs.oracle.com/en/java/
- StarUML – staruml.io | draw.io – draw.io | GitHub Copilot – github.com/features/copilot

Course Code	DIGITAL LOGIC AND COMPUTER ORGANIZATION LAB	L	T	Pr	P	Credits
26AEC04301P	AI & DS | II Year I Sem	0	0	0	2	1
Prerequisite:	Digital Logic and Computer Organization (co-requisite)					

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

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)**

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.

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.

LEARNING RESOURCES

Textbooks

1. Hamacher, C., Vranesic, Z., & Zaky, S. (2023). Computer Organization (6th ed.). McGraw Hill.
2. Mano, M. M. (2018). Digital Design (6th ed.). Pearson Education.

Reference Books

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

Online Resources

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

Course Code	ALGORITHMS DESIGN AND ANALYSIS LAB	L	T	Pr	P	Credits
26ACS05301P	Common to CSE and Allied Branches	0	0	0	3	1.5

Lab Course Outcomes

- Implement and analyse sorting and searching algorithms and verify time complexity empirically using Python.
- Implement divide-and-conquer algorithms and compare their performance with brute-force alternatives.
- Implement greedy algorithms for MST, shortest path, scheduling, and Huffman coding.
- Implement dynamic programming solutions for classic problems and compare memoization vs. tabulation.
- Implement backtracking, branch-and-bound, and approximation algorithms; classify problem complexity experimentally.

List of Experiments

Unit I | Algorithm Analysis and Mathematical Foundations

1. **Complexity Visualization and Growth Rate Comparison**
   Implement Linear Search, Binary Search, Bubble Sort, Merge Sort, and Quick Sort in Python. Measure execution time for each on arrays of size n = 1,000 to 1,000,000 using the time module. Plot the measured execution times vs. n alongside the theoretical O(n), O(n log n), O(n^2), and O(log n) curves using Matplotlib. Verify experimentally that the empirical curves match the predicted asymptotic growth. Use VisuAlgo to animate each algorithm side by side.

2. **Recurrence Solving and Master Theorem Verification**
   Implement Merge Sort and Binary Search recursively in Python. Instrument each recursive call to count the total number of recursive calls made for a given input size n. For Merge Sort ($T(n) = 2T(n/2) + O(n)$) and Binary Search ($T(n) = T(n/2) + O(1)$), apply the Master theorem to derive the closed-form solution. Plot the measured call count vs. n and the Master theorem prediction on the same graph. Use SymPy to verify the recurrence solution symbolically.

3. **Amortized Analysis — Dynamic Array and Stack**
   Implement a dynamic array in Python that doubles its capacity on overflow (similar to Python list internals). Insert 10,000 elements and record the actual cost of each insertion (1 for normal, n for resize). Plot the cumulative cost vs. number of insertions. Show using the aggregate method that amortized cost per operation is O(1). Repeat for a stack with multi-pop operations and verify amortized cost using the accounting method.

Unit II | Divide and Conquer

4. **Sorting Algorithm Comparison — Merge Sort, Quick Sort, Heap Sort**
   Implement Merge Sort, Quick Sort (with three pivot strategies: first element, random, median-of-three), and Heap Sort in Python. Measure execution time on: (a) random arrays, (b) already-sorted arrays, (c) reverse-sorted arrays of size 1,000 to 100,000. Plot a grouped bar chart comparing all strategies across all three input types. Verify that worst-case Quick Sort (sorted input, first pivot) degrades to O(n^2) and that random pivot avoids this. Animate each sort on a 20-element array using VisuAlgo.

5. Randomized Selection and Order Statistics

Implement the Randomized Select algorithm in Python to find the k-th smallest element in an unsorted array in expected O(n) time. Also implement the deterministic O(n) Median-of-Medians algorithm. Compare execution time of both against Python's built-in sort-then-index (O(n log n)) for k = n/2 (median) on arrays of size 10,000 to 500,000. Plot execution time vs. n for all three methods. Verify empirically that Randomized Select runs in expected O(n).

6. Strassen's Matrix Multiplication

Implement classical $O(n^3)$ matrix multiplication and Strassen's O(n^2.807) algorithm in Python. Measure execution time for n × n matrices with n = 2, 4, 8, 16, 32, 64, 128, 256. Plot execution time vs. n on a log-log scale and verify that the slope difference corresponds to the complexity ratio. Identify the crossover point (the n at which Strassen outperforms classical multiplication on your machine). Discuss the cache performance implications.

Unit III | Greedy Algorithms

7. Activity Selection and Job Sequencing

Implement the Activity Selection problem using a greedy approach (sort by finish time, select non-overlapping activities). Generate 5 test cases with n = 10, 20, 50 activities and verify that the greedy solution is optimal by comparing with exhaustive search for $n \leq 15$. Implement Job Sequencing with Deadlines: given n jobs each with a deadline and profit, find the maximum-profit subset that can be completed before respective deadlines. Plot profit vs. number of jobs selected for varying inputs.

8. Minimum Spanning Tree — Prim's and Kruskal's

Implement Prim's algorithm using a priority queue (heapq) and Kruskal's algorithm using Union-Find (path compression + union by rank) in Python. Apply both to 5 weighted graphs of increasing size (10 to 500 nodes) generated using NetworkX. Verify that both produce the same MST weight. Measure and plot execution time vs. number of edges for both algorithms. Visualize the MST using NetworkX and Matplotlib, coloring MST edges red on the full graph.

9. Dijkstra's Shortest Path and Huffman Coding

Implement Dijkstra's shortest path algorithm using a min-heap on a weighted directed graph. Find shortest paths from a source to all nodes on 3 test graphs (sparse, dense, random) of 100 nodes. Verify correctness against NetworkX's built-in shortest path. Separately, implement Huffman coding: given a string of at least 100 characters, build the Huffman tree, generate codes, encode the string, and compute the compression ratio. Compare the compressed bit length against ASCII fixed-width (8 bits/char) and verify the prefix-free property.

Unit IV | Dynamic Programming

10. Longest Common Subsequence and Edit Distance

Implement the LCS algorithm in Python using both memoization (top-down with @functools.lru_cache) and tabulation (bottom-up 2D table). Print the DP table and trace back the actual LCS string for 5 string pairs. Implement the Edit Distance (Levenshtein) algorithm for the same pairs. Print the edit distance table and recover the sequence of insert, delete, and substitute operations. Measure and compare execution time and memory usage of memoization vs. tabulation for strings of length 100 to 1,000.

11. 0/1 Knapsack and Subset Sum

Implement the 0/1 Knapsack problem using tabulation and memoization in Python. Given n items with weights and values (n = 10 to 50) and a knapsack capacity W, compute the maximum value and print the selected items by tracing back the DP table. Implement Subset Sum as a special case (value = weight, target = W). For $n \leq 20$, verify the DP result against an exhaustive search (2^n subsets). Plot DP table fill time vs. n × W to demonstrate pseudo-polynomial complexity.

12. Floyd-Warshall All-Pairs Shortest Path

Implement the Floyd-Warshall algorithm in Python for all-pairs shortest path on a weighted directed graph (with possible negative edges but no negative cycles). Run on 3 test graphs of 10, 50, and 100 nodes. Print the distance matrix after every k-th iteration (k = 1, 5, 10) to visualize how paths are refined. Detect and report negative cycles. Compare execution time with running Dijkstra from each source node (for non-negative graphs). Visualize the shortest path between a selected source-destination pair using NetworkX.

Unit V	Advanced Topics and Computational Complexity

13. Backtracking — N-Queens and Graph Coloring

Implement the N-Queens problem using backtracking in Python. Solve for N = 4, 6, 8, 10, 12 and count the number of solutions and the number of nodes explored in the state space tree. Plot nodes explored vs. N. Visualize the solution board using Matplotlib. Implement Graph Coloring using backtracking: given a graph and k colors, find a valid coloring (or report none exists). Test on 5 graphs of 10–20 nodes. Visualize the colored graph using NetworkX with node colors reflecting the assigned color.

14. Branch and Bound — TSP

Implement the Branch and Bound algorithm for the Travelling Salesman Problem (TSP) in Python using a lower bound based on reduced cost matrices. Solve for complete graphs of n = 5, 8, 10, 12 cities with random edge weights. Compare the optimal tour cost from Branch and Bound with: (a) nearest-neighbour greedy heuristic, (b) Christofides-style 2-approximation. Record and plot solution quality (tour cost) and computation time vs. n for all three approaches. Document the point at which Branch and Bound becomes infeasible within 60 seconds.

15. NP-Completeness Verification and Approximation Algorithms

Implement the 2-approximation algorithm for Vertex Cover (repeatedly pick an uncovered edge, add both endpoints) and the greedy Set Cover algorithm in Python. Verify 2-approximation: for 10 random graphs of 20–50 nodes, compare the approximate vertex cover size vs. the optimal cover (computed by brute force for small graphs). Plot approximation ratio vs. n. Separately, implement Randomized QuickSort and Las Vegas random selection. Run each 100 times on the same input and plot the distribution of comparisons made, verifying expected O(n log n) and O(n) behaviour respectively.

AI Tools Integration (Lab)	Use Google Colab as the primary lab environment — no local setup required. Use VisuAlgo (visualgo.net) to animate sorting, graph traversal, and DP table construction before implementing. Use Python Tutor to trace recursive call stacks for Merge Sort, Quick Sort, and DP memoization. Use NetworkX and Matplotlib for graph construction and visualization of MST, shortest paths, and colored graphs. Use SymPy to verify recurrence solutions and Wolfram Alpha to cross-check asymptotic results. Use ChatGPT or Claude to explain algorithm steps, verify DP table traces, and debug implementation errors — all AI-generated code must be tested and documented before submission.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L3	Apply	**CO1, CO2**	Implement sorting, searching, and divide-and-conquer algorithms in Python.
L4	Analyse	**CO2, CO3**	Measure empirical complexity; trace graph algorithms; compare dynamic programming approaches.

L5	Evaluate	**CO4, CO5**	Evaluate algorithm selection per problem; assess NP-completeness reductions.
L6	Create	**CO3, CO5**	Design a novel greedy or DP solution 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 PrjMgm
CO1	L3	3	3	1	3	2	–	–	–	–	1	–
CO2	L4	3	3	2	3	2	–	–	–	–	1	–
CO3	L4	3	3	3	3	3	–	–	–	1	2	–
CO4	L5	3	3	3	3	3	–	–	–	1	2	1
CO5	L5	3	3	3	3	3	1	–	–	1	2	1

CO-PO Correlation Highlights	Lab experiments directly reinforce PO1 and PO4 (Investigation) through complexity measurement and empirical benchmarking. PO3 (Design/Development) through capstone algorithm design experiments. PO5 through VisuAlgo, Gephi, NetworkX, Python Tutor, and Google Colab across all 15 experiments.

LEARNING RESOURCES

Textbooks

1. TB1: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
2. TB2: Jon Kleinberg and Eva Tardos, Algorithm Design, Pearson Education, 2013.

Reference Books

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, Algorithms, McGraw-Hill, 2008. (Free PDF at algorithms.wtf)
2. Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011.

Online Resources

- Google Colab – colab.research.google.com | Python Tutor – pythontutor.com
- VisuAlgo – visualgo.net | Algorithm Visualizer – algorithm-visualizer.org
- NetworkX – networkx.org | Matplotlib – matplotlib.org |SymPy – sympy.org
- NPTEL – Design and Analysis of Algorithms, IIT Madras – nptel.ac.in
- MIT OpenCourseWare – Introduction to Algorithms (6.006) – ocw.mit.edu
-

Course Code	OBJECT ORIENTED DESIGN AND PROGRAMMING LAB	L	T	Pr	P	Credits
26ACS05302P	Common to CSE and Allied Branches	0	0	0	3	1.5

Lab Course Outcomes

- Write and trace Java programs using data types, operators, control statements, and OOP principles.
- Design and implement classes with constructors, overloaded methods, access control, and Java Collections.
- Implement inheritance hierarchies, interfaces, abstract classes, and dynamic dispatch in Java.
- Design multi-package applications with exception handling, file I/O, and Java standard library APIs.
- Build multithreaded programs with synchronization, connect Java to MySQL using JDBC, and develop a JavaFX GUI application.

List of Experiments

Unit I | OOP Concepts, Data Types, Operators, and Control Statements

1. Java Basics — Data Types, Operators, and Type Casting

Write Java programs to: (a) declare and print variables of all primitive data types (byte, short, int, long, float, double, char, boolean) with their storage sizes using sizeof equivalent; (b) demonstrate implicit and explicit type casting — int to double, double to int, char to int; (c) demonstrate all operator categories — arithmetic, relational, logical, bitwise (AND, OR, XOR, NOT, left shift, right shift), assignment, ternary; (d) print the operator precedence table and write 5 expressions with mixed operators, predict the result manually, then verify by running the program. Use GitHub Copilot to generate a test harness for each operator category and validate all edge cases.

2. Control Statements — Decision Making and Loops

Write Java programs to: (a) implement a menu-driven calculator using switch that supports +, −, ×, ÷, %, and power operations; (b) print all prime numbers between 1 and 1,000 using a for loop with break; (c) print the following patterns using nested loops — right-angled triangle, inverted triangle, diamond, Pascal's triangle; (d) implement a number guessing game using a while loop — the program generates a random number (1–100) and gives the user a maximum of 7 attempts with higher/lower hints. Use Python Tutor (Java mode) or draw.io to trace and visualize the control flow of each program before coding.

3. Command-Line Arguments and Static Members

Write Java programs to: (a) accept student name, roll number, and 5 subject marks as command-line arguments, compute total and percentage, and display the grade (O/A+/A/B+/B/C/F) using if-elif-else; (b) implement a class Counter with a static variable count that tracks the total number of Counter objects created across all instances — print the count from both the instance and class level; (c) demonstrate the difference between static and non-static methods by implementing a MathUtils class with static utility methods (GCD, LCM, isPrime, factorial) called without creating an object. Design a UML class diagram for each using StarUML before implementation.

Unit II | Classes, Objects, and Methods

4. Class Design — Student Management System

Design and implement a Student class with fields: rollNo (int), name (String), marks[] (double array for 5 subjects), and a calculated grade (char). Include: (a) a default constructor, a parameterized constructor, and a copy constructor; (b) overloaded constructors — one accepting only rollNo and name, one accepting all fields; (c) a computeGrade() method using the percentage; (d) overloaded display() methods

— one prints summary (name and grade), one prints full details; (e) a static method topStudent(Student[] s) that returns the student with the highest percentage. Draw the UML class diagram in StarUML before coding. Create an array of 5 Student objects and demonstrate all methods.

5. **Method Overloading, Recursion, and this Keyword**
   Write a Java class Calculator with overloaded add() methods that accept: (a) two integers; (b) three integers; (c) two doubles; (d) an integer array (sum of all elements). Demonstrate that Java resolves the correct method at compile time (static dispatch). Separately, implement a RecursionDemo class with recursive methods for: factorial (both recursive and iterative — compare for n = 1 to 20), Fibonacci (recursive — trace the call tree for n = 8 using Python Tutor), binary search on a sorted array, and power(base, exp) using divide and conquer. Demonstrate the this keyword: use this() to call one constructor from another (constructor chaining) and this.field to resolve name shadowing.

6. **Java Collections — ArrayList, HashMap, and TreeMap**
   Write programs demonstrating: (a) ArrayList — store 10 student names, sort alphabetically (Collections.sort), search using indexOf, remove duplicates using a LinkedHashSet, and iterate using for-each and iterator; (b) HashMap — build a word frequency counter that reads a sentence, splits into words, and stores word → count pairs; print the top 5 most frequent words; (c) TreeMap — store student roll numbers and marks in a TreeMap (auto-sorted by key), iterate in ascending and descending order (descendingKeySet()), and find the student with the highest and lowest marks using firstKey() and lastKey(). Compare ArrayList vs. LinkedList on insertion-at-head time for 100,000 elements.

Unit III | Arrays, Inheritance, and Interfaces

7. **Arrays — 1D, 2D, Sorting, and Searching**
   Write Java programs for: (a) 1D array operations — insert at position, delete by value, reverse in place, find second largest, rotate left by k positions; (b) 2D array operations — matrix addition, matrix multiplication, transpose, find row with maximum sum; (c) sorting — implement Bubble Sort, Selection Sort, and Insertion Sort; compare their execution time on an array of 10,000 random integers; verify result against Arrays.sort(); (d) searching — implement linear search and binary search (iterative and recursive); for binary search, count the number of comparisons for 20 random queries on a sorted array of 1,000 elements and plot comparisons vs. element position.

8. **Inheritance — Vehicle Hierarchy and Runtime Polymorphism**
   Design and implement a multilevel inheritance hierarchy: Vehicle (base) → MotorVehicle → Car and Truck (hierarchical from MotorVehicle). Vehicle has: make (String), model (String), year (int), fuelType (String); abstract method fuelEfficiency(). Car adds: seatingCapacity (int); Truck adds: payloadCapacity (double). Override fuelEfficiency() differently in Car and Truck. Demonstrate: (a) constructor chaining using super() — print the constructor call order; (b) method overriding — call the same method on a Vehicle reference pointing to Car and Truck objects and verify dynamic dispatch (runtime polymorphism); (c) final method and final class — make a method in MotorVehicle final and verify that Car cannot override it. Draw the full UML hierarchy in StarUML before coding.

9. **Interfaces — Payment System and Functional Interfaces**
   Design a payment processing system using interfaces: (a) Payable interface with processPayment(double amount), getReceipt(), and refund(double amount) methods; implement CreditCardPayment, UPIPayment, and NetBankingPayment — each with different processing logic; demonstrate multiple interface implementation by creating a PremiumPayment class that implements both Payable and Auditable (a second interface with logTransaction() method); (b) demonstrate default methods in interfaces — add a default validateAmount(double amount) method to Payable that checks if amount > 0; show that implementing classes inherit the default without overriding; (c) implement a functional interface MathOperation with a single abstract method operate(int a, int b) — instantiate it using three lambda expressions: add, subtract, multiply.

Unit IV | Packages, Exception Handling, and Java I/O

10. Packages and Java Standard Library APIs

Create a multi-package application: package com.university.model (Student, Course classes), package com.university.utils (GradeCalculator with static methods), package com.university.main (Main class). Demonstrate access control: show that a private field in Student is not accessible from com.university.main directly. In the same program, demonstrate Java.lang APIs: (a) Math class — compute sin, cos, log, sqrt, pow, and floor for 5 values; (b) Wrapper classes — parse int from String (Integer.parseInt), convert int to binary/octal/hex strings (Integer.toBinaryString), and demonstrate auto-boxing/unboxing in an ArrayList<Integer>; (c) java.time API — get the current date and time (LocalDateTime.now()), add 30 days (plusDays), find the day of the week, and format using DateTimeFormatter.

11. Exception Handling — Robust Bank Account Application

Implement a BankAccount class that throws custom exceptions: (a) InsufficientFundsException (checked) — thrown when withdrawal exceeds balance; (b) InvalidAmountException (unchecked) — thrown when a negative or zero amount is passed to deposit or withdraw; (c) AccountFrozenException (checked) — thrown when a frozen account is accessed. Implement deposit(), withdraw(), and transfer() methods that throw these exceptions appropriately. Write a BankingApp with a menu-driven interface that: catches each exception with a separate catch block; uses a finally block to log every transaction attempt (successful or failed) to a log file; uses try-with-resources to auto-close the log file writer. Test with at least 10 scenarios covering all three exception types.

12. Java File I/O — Student Record File System

Build a file-based student record system using Java I/O: (a) write student records (roll number, name, 5 subject marks) to a text file using BufferedWriter and read them back using BufferedReader — parse each line and reconstruct Student objects; (b) read a large text file (at least 500 lines) using Java NIO (Files.readAllLines and Files.lines with Stream API) — count total words, find the longest line, and count lines containing a specific keyword; (c) copy a binary file (any image) using FileInputStream and FileOutputStream in chunks of 4KB — measure and report the time taken; (d) demonstrate RandomAccessFile — write 10 student records, then directly seek to and update the 5th record without rewriting the entire file.

Unit V | Strings, Multithreading, JDBC, and JavaFX

13. String Handling and StringBuffer

Write Java programs demonstrating: (a) String class methods — length, charAt, indexOf, lastIndexOf, substring, replace, toUpperCase, toLowerCase, trim, split, contains, startsWith, endsWith — apply each to at least 3 test strings; (b) String immutability — demonstrate that concatenating 10,000 strings using + operator in a loop is slow; replace with StringBuilder and measure the speedup; explain why String is immutable in Java (security, thread safety, string pool); (c) StringBuffer vs. StringBuilder — implement a palindrome checker, a word reverser, and an anagram checker using StringBuffer; (d) CharSequence interface — write a method that accepts a CharSequence parameter and works correctly when passed a String, StringBuilder, and StringBuffer.

14. Multithreading — Producer-Consumer and Synchronization

Implement the following multithreaded programs: (a) create 5 threads using both Thread class extension and Runnable interface implementation — name each thread and print its name and ID; demonstrate thread priority (set priorities 1–10 and observe scheduling); (b) implement the Producer-Consumer problem using a shared bounded buffer (size = 5) — producer adds items when buffer is not full, consumer removes items when not empty; use synchronized methods with wait() and notifyAll() to coordinate; run 2 producer threads and 3 consumer threads simultaneously; (c) demonstrate a deadlock — two threads each hold one lock and wait for the other; detect it using thread dump (jstack); then resolve it by enforcing a consistent lock ordering; (d) implement a thread-safe bank account with synchronized

deposit() and withdraw() methods — run 10 threads performing simultaneous operations and verify the final balance is correct.

15. **JDBC and JavaFX — Student Record Management Application**
Build a complete Student Record Management application: (a) JDBC backend — create a MySQL database with a students table (roll_no INT PRIMARY KEY, name VARCHAR, dept VARCHAR, cgpa DECIMAL); implement a StudentDAO class with insertStudent(), getAllStudents(), updateCGPA(rollNo, newCGPA), and deleteStudent(rollNo) using PreparedStatement for all queries (no Statement — prevent SQL injection); demonstrate CRUD by inserting 5 students, displaying all, updating 2, and deleting 1; (b) JavaFX frontend — build a GUI with: a TableView displaying all students (fetched from MySQL via JDBC), a form panel with TextField inputs for roll_no, name, dept, cgpa and Add/Update/Delete buttons, and a Search field that filters the TableView in real time; wire all buttons to the DAO methods; (c) integrate exception handling throughout — catch SQLException and show an Alert dialog to the user on any database error.

AI Tools Integration (Lab)	Use GitHub Copilot in IntelliJ IDEA or Eclipse for code generation and intelligent autocomplete — always review, test, and understand the generated code before submission. Use ChatGPT or Claude to explain error messages, debug NullPointerExceptions and thread deadlocks, and verify logic for inheritance hierarchies and JDBC queries. Use StarUML to design UML class and sequence diagrams before each experiment (design-first approach). Use draw.io to create flowcharts and object interaction diagrams for control flow and method call tracing. Use Python Tutor (Java mode) to visualize recursive call stacks and inheritance constructor chaining. Use Scene Builder for drag-and-drop JavaFX UI design in Experiment 15.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L3	Apply	**CO1, CO5**	Execute Java programs for control flow; implement multithreaded producer-consumer.
L4	Analyse	**CO2–CO4**	Design class hierarchies; analyse exception propagation; compare serialisation formats.
L5	Evaluate	**CO3, CO4**	Evaluate interface vs. abstract class; assess collection choice for performance.
L6	Create	**CO5**	Build a complete JavaFX + JDBC Student Management Application (Experiment 15).

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	1	1	2	–	–	–	–	1	–
CO2	L4	3	3	3	2	2	–	–	–	1	1	–
CO3	L4	3	3	3	2	2	–	–	–	1	1	–

CO4	L4	3	3	2	2	2	–	–	–	–	1	–
CO5	L3	3	2	2	1	2	–	–	–	1	1	–

CO-PO Correlation Highlights	Strong PO1 through Java fundamentals and OOP implementation. PO3 (Design/Development) through class design, inheritance, and JavaFX capstone application. PO5 through StarUML, GitHub Copilot, IntelliJ IDEA, Scene Builder, MySQL Workbench, and JDBC across all experiments.

LEARNING RESOURCES

Textbooks

1. TB1: Anitha Seth and B.L. Juneja, JAVA one step ahead, Oxford University Press.
2. TB2: Debasis Samanta and Monalisa Sarma, Joy with JAVA — Fundamentals of Object Oriented Programming, Cambridge University Press, 2023.
3. TB3: Paul Deitel and Harvey Deitel, Java 9 for Programmers, 4th Edition, Pearson Education.

Reference Books

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

Online Resources

- Oracle Java Documentation – docs.oracle.com/en/java/
- NPTEL – Programming in Java – nptel.ac.in/courses/106/105/106105191/
- StarUML – staruml.io | draw.io – draw.io | Scene Builder – gluonhq.com/products/scene-builder
- Python Tutor (Java mode) – pythontutor.com | GitHub Copilot – github.com/features/copilot

Course Code	SUSTAINABILITY AND GREEN ENGINEERING	L	T	Pr	P	Credits
26AHS00305T	AI & DS | II Year I Sem	2	0	0	0	2
Prerequisite:	None					

Prerequisites: None

Course Objectives

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

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**

Course Content

UNIT – 1	Fundamentals of Sustainability and Carbon Cycle	6 Hrs
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 – 2	Sustainable Materials and Resource Selection	6 Hrs
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 – 3	Embodied Energy and Operational Energy	6 Hrs
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 – 4	Green Rating Systems and Energy Codes	6 Hrs
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 – 5	Circular Economy and Environmental Sustainability	6 Hrs
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.	

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 (NBA)

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	–	–	–	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 Highlights	CO4 and CO5 show strong correlation with PO3 (Design) and PO5 (Modern Tools) through green rating compliance and AI-driven 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.

LEARNING RESOURCES

Textbooks

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

Reference Books

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

Online Resources

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

Web Resources & NPTEL Links

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

Recommended Free & Open-Source AI/Computational Tools
openLCA (free, open-source) – Life Cycle Assessment software for embodied carbon calculation
EnergyPlus (free, US DOE) – Building energy simulation for operational energy analysis OpenStudio (free, NREL) – Building energy modeling interface for EnergyPlus Python (scikit-learn, Pandas, NumPy) on Google Colab – ML for energy prediction and material selection
QGIS (free) – Green infrastructure planning and spatial sustainability analysis i-Tree Tools (free, USDA) – Urban tree carbon sequestration quantification
SimaPro (student trial) – Life cycle assessment platform

Course Code	AI ASSISTED APPLICATION DEVELOPMENT	L	T	Pr	P	Credits
26AAI30301S	Common to CSE and Allied Branches	1	0	0	2	2

Course Objectives

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

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)

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 Content

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 (npminit), 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.

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.	PO11 PrjMgm
CO1	L6	3	2	3	1	3	2	–	1	1	2	1

CO2	L6	3	3	3	2	3	1	–	1	1	2	1
CO3	L6	3	3	3	2	3	1	–	1	1	2	1
CO4	L6	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 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

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

Course Code	TECHNICAL REPORT WRITING & INTELLECTUAL PROPERTY RIGHTS	L	T	Pr	P	Credits
26AHS00302M	AI & DS | II Year II Sem	2	0	0	0	0
Prerequisite:	Communicative English					

Prerequisites: Communicative English

Course Objectives

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

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)**

Course Content

UNIT – 1	Foundations of Technical Communication	6 Hrs
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 – 2	**Technical Report Writing**	**6 Hrs**
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 – 3	**Research Proposals and Scientific Writing**	**6 Hrs**
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 – 4	**Intellectual Property Rights — Fundamentals**	**6 Hrs**
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 – 5	**Patent Drafting and IP Strategy**	**6 Hrs**
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.

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	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	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.

LEARNING RESOURCES

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 Allied Rights (9th ed.). Sweet & Maxwell.
5. World Intellectual Property Organization (WIPO). (2022). WIPO Intellectual Property Handbook. WIPO Geneva.

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

Course Code	MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS	L	T	Pr	P	Credits
26AMB20301T	AI & DS | II Year II Sem	2	0	0	0	2
Prerequisite:	None					

Prerequisites: None

Course Objectives

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

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 fundamental concepts of economics including demand, supply, market structures, and the time value of money.	**L1–L2**
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.	**L5 (Evaluate)**
CO5	Create a comprehensive mini-project business plan incorporating market analysis, cost estimation, and AI-assisted forecasting.	**L6 (Create)**

Course Content

UNIT – 1	Introduction to Managerial Economics	6 Hrs
Content	• Managerial Economics: Meaning, Nature, Scope and significance for Engineers; Economic Analysis — Micro vs. Macroeconomics; positive vs. normative economics. • Demand Analysis: Concept, Determinants, Demand Function, Law of Demand; Elasticity of Demand: Types, Measurement, Business Applications; Demand Forecasting: Meaning, Need, Methods.	

AI Integration	AI Tool: ChatGPT to generate demand scenarios for an engineering product and interpret price elasticity. AI Tool: Google Sheets + FORECAST function for Moving Average and Exponential Smoothing. Activity: AI-assisted regression demand forecasting in Google Colab.

UNIT – 2	**Production and Cost Analysis**	**6 Hrs**
Content	• Production Concepts and Modern Manufacturing Systems; Production Function: Isoquants and Isocosts; Least-Cost Combination of Inputs. • Cost Concepts: Fixed, Variable, Total, Average, Marginal Costs; Break-Even Analysis: Concept, Assumptions, Applications; Margin of Safety and Profit Planning.	
AI Integration	AI Tool: Google Sheets/LibreOffice Calc for BEA spreadsheet with dynamic charts. AI Tool: ChatGPT to generate cost functions and identify optimal production level. Activity: AI-assisted CVP analysis using Plotly for cost curve visualisation.	

UNIT – 3	**Business Organisations and Market Structures**	**6 Hrs**
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 with real-world examples; Pricing Methods and Strategies.	
AI Integration	AI Tool: ChatGPT to simulate a duopoly pricing game and find Nash Equilibrium. Activity: analyse pricing strategies of competing tech products. Discussion: AI-driven dynamic pricing in e-commerce (Amazon, Uber Surge Pricing).	

UNIT – 4	**Capital Budgeting**	**6 Hrs**
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 to evaluate capital investment alternatives. Monte Carlo simulation for capital budgeting risk analysis. Decision tree analysis for investment appraisal using scikit-learn.	

UNIT – 5	**Financial Accounting and Analysis**	**6 Hrs**
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 to interpret company financial ratios. Project: ratio analysis of an Indian engineering company's annual report (BHEL, L&T, Infosys).	

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L1/L2	Remember/Understand	**CO1**	Demand and economics concept recall; market structure identification.
L3	Apply	**CO2**	Break-even and cost function application exercises.
L4	Analyse	**CO3**	Financial statement and ratio analysis using AI tools.
L5/L6	Evaluate/Create	**CO4, CO5**	NPV/IRR investment evaluation; business plan mini-project 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	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

CO-PO Correlation Highlights	CO3 and CO4 align strongly with PO2 (Problem Analysis), PO4 (Investigation), and PO5 (Modern Tools) through AI-powered financial analysis and investment evaluation. CO5 shows the broadest correlation, mapping to PO3, PO9, PO10, and PO11 through the integrative business plan capstone project. PO11 (Project Management) is consistently strong given the financial decision-making focus of the course.

LEARNING RESOURCES

Textbooks

1. Mithani, D. M. (2025). Managerial Economics: Theory and Applications (10th ed.). Himalaya Publishing House.
2. Pandey, I. M. (2022). Financial Management (12th ed.). Vikas Publishing House.

Reference Books

1. Aryasri, A. R. (2020). Managerial Economics and Financial Analysis (Revised ed.). McGraw-Hill Education.
2. Prasanna Chandra (2019). Financial Management: Theory and Practice (10th ed.). Tata McGraw-Hill.
3. Kumar, U. D. (2021). Business Analytics: The Science of Data-Driven Decision Making (2nd ed.). Wiley India.
4. Siddiqui, S. A. (2019). Engineering Economics and Financial Accounting (3rd ed.). S. Chand Publishing.

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 – https://www.coursera.org/learn/economics
- Khan Academy: Microeconomics & Finance Basics

SWAYAM Portal: Business Economics – https://swayam.gov.in

Online Resources:

S.No.	Platform	Course / Resource	Link / Access
1	NPTEL (IIT)	Managerial Economics	https://nptel.ac.in/courses/110105063
2	NPTEL (IIT)	Financial Accounting for Engineers	https://nptel.ac.in/courses/110107114
3	Coursera	Economics for Non-Economists (Univ. of California)	https://www.coursera.org/learn/economics
4	edX	Financial Accounting (MIT OpenCourseWare)	https://www.edx.org/course/financial-accounting
5	Khan Academy	Microeconomics & Finance Basics	https://www.khanacademy.org/economics-finance-domain
6	SWAYAM Portal	Business Economics (UGC-approved)	https://swayam.gov.in

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, spreadsheet 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

AI-INTEGRATED ACTIVITIES

Unit	Activity	Description	AI Tool Used	Outcome Mapped
I	Demand Forecasting Simulation	Use AI tools to forecast demand for a consumer product using historical data	ChatGPT + Statista	CO1
I	Elasticity Chatbot Quiz	AI-generated adaptive quiz on types and measurement of elasticity	Google Gemini	CO1

II	Break-Even Calculator	Build a Break-Even model using Excel + Copilot for sensitivity analysis	Microsoft Copilot	CO2
II	Cost Curve Visualisation	Plot and interpret cost curves using AI-assisted data tools	Wolfram Alpha / Tableau	CO2
III	Market Structure Case Study	Generate real-world case studies of oligopoly (e.g., Telecom sector) using AI	Claude AI / ChatGPT	CO3
III	Startup Evaluation Report	Evaluate a digital startup's business model using AI-generated insights	Google Gemini	CO3
IV	Capital Budgeting Decision Tool	Use AI to compare NPV, IRR for two competing engineering projects	Copilot + Wolfram Alpha	CO4
IV	IRR Sensitivity Analysis	Explore how changes in cash flows affect IRR using AI-powered spreadsheets	Microsoft Copilot	CO4
V	Digital Accounting Exercise	Prepare journal, ledger, and trial balance using Tally AI / Zoho Books	Tally AI / Zoho AI	CO5
V	Financial Ratio Dashboard	Construct a financial ratio dashboard from a company's annual report using Tableau AI	Tableau AI	CO5

Course Code	UNIVERSAL HUMAN VALUES – UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT	L	T	Pr	P	Credits
26AHS00302T	AI & DS | II Year I Sem	3	0	0	0	3
Prerequisite:	None					

Prerequisites: None

Course Objectives

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

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)**

Course Content

UNIT – 1	Introduction to Value Education and Human Consciousness	9 Hrs
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 – 2	Harmony in the Individual	9 Hrs
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 – 3	Harmony in the Family and Society	9 Hrs
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 – 4	Harmony in Nature and Existence	9 Hrs
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 – 5	Ethical Human Conduct and Professional Ethics	9 Hrs
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.	

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	PSO1	PSO2
CO1	–	–	–	–	–	2	2	3	1	2	–	2	1	1
CO2	–	–	–	–	–	3	2	3	2	2	1	2	1	1
CO3	1	2	–	1	–	3	2	3	2	2	1	3	1	2
CO4	1	2	2	2	2	3	3	3	2	2	2	3	2	2
CO5	1	2	2	1	2	3	2	3	3	3	2	3	2	2

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.

LEARNING RESOURCES

Textbooks

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

Reference Books

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

Online Resources

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

YouTube / Video Resources

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

Course Code	INTELLIGENT DATA PREPARATION AND FEATURE ENGINEERING	L	T	Pr	P	Credits
26AAI30401T	For AI& DS Branch	2	0	2	0	3
Prerequisite	Data Structures; Discrete Mathematics; Python Programming					

Course Objectives

- Develop systematic understanding of data collection, exploration, and profiling as the foundation of every AI and data science workflow.
- Build competency in data cleaning, preprocessing, class imbalance handling, and data integration for real-world datasets.
- Develop knowledge of feature engineering for structured data across numerical, categorical, datetime, and text column types.
- Introduce feature selection and dimensionality reduction as formal processes for identifying informative data representations.
- Expose students to automated feature engineering, feature pipelines, and responsible feature engineering practices.

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 data collection methods, EDA techniques, and data quality dimensions; assess a dataset for AI readiness. (L2)	L2 (Understand)
CO2	Classify missing data patterns and apply appropriate cleaning, preprocessing, and integration strategies. (L4)	L4 (Analyse)
CO3	Design and justify feature engineering strategies for numerical, categorical, datetime, and text data for a given prediction task. (L4)	L4 (Analyse)
CO4	Apply and compare filter, wrapper, and embedded feature selection methods; evaluate PCA, LDA, t-SNE, and UMAP for a given task. (L4)	L5 (Evaluate)
CO5	Describe automated feature engineering, feature pipeline design, and responsible feature engineering practices. (L4)	L2 (Understand)

Course Content

UNIT – I	DATA COLLECTION, EXPLORATION, AND PROFILING	9 Hrs

Content	
	- CRISP-DM workflow; data source types: structured (CSV, SQL), semi-structured (JSON, XML), unstructured (text, images); REST APIs and web scraping concepts; public datasets — UCI ML Repository, Kaggle, data.gov.in; collection biases: selection, survivorship, response, temporal. - Exploratory Data Analysis (EDA): univariate analysis (histograms, box plots, bar charts); bivariate analysis (scatter plots, correlation — Pearson, Spearman, Kendall); multivariate analysis (pair plots, heatmaps); distribution analysis (normal, log-normal, Poisson); time-series EDA (trend, seasonality, ACF, PACF).

	□ Data quality dimensions: accuracy, completeness, consistency, timeliness, validity, uniqueness; data profiling outputs (null rates, cardinality, data type mismatches); AI-readiness checklist; automated profiling tools — ydata-profiling, Great Expectations. **Source:** TB1: Chapter 2 (Géron — end-to-end ML project, EDA); TB2: Chapters 1–3 (McKinney — data loading, cleaning overview, visualisation)
AI Integration	Use ChatGPT to explain CRISP-DM phases with a concrete dataset example; use Claude to generate an EDA plan for a given dataset description and evaluate it; use an AI assistant to generate a data quality checklist across all six dimensions.

UNIT – II	DATA CLEANING, PREPROCESSING, AND INTEGRATION	9 Hrs

Content	□ Missing data: MCAR, MAR, MNAR — formal definitions; deletion strategies (listwise, column deletion); univariate imputation (mean, median, mode, constant); multivariate imputation (KNN imputation, MICE); indicator variables for missingness. □ Outliers and noise: types (statistical, contextual, collective); detection — Z-score, modified Z-score, IQR, Mahalanobis distance, Isolation Forest, LOF; treatment — removal, Winsorisation, transformation; noise smoothing — moving average, EWMA. □ Class imbalance: evaluation metrics (precision, recall, F1, PR-AUC, ROC-AUC); oversampling — SMOTE, ADASYN; undersampling — Tomek links, ENN; algorithm-level — class_weight='balanced', cost-sensitive learning. □ Data integration and transformation: schema mapping, entity resolution, record linkage; normalisation (min-max, z-score, robust scaling); discretisation (equal-width, equal-frequency, entropy-based); data type conversions. **Source:** TB2: Chapters 5–7 (McKinney — missing data, merging, grouping); TB3: Chapters 3–5 (Kuhn & Johnson — preprocessing, missing values, class imbalance)
AI Integration	Use ChatGPT to explain MICE step by step for a dataset with missing values; use Claude to compare SMOTE and ADASYN for a given class imbalance scenario; use an AI assistant to explain Isolation Forest anomaly detection on a 2D example.

UNIT – III	FEATURE ENGINEERING FOR STRUCTURED DATA	9 Hrs

Content	□ Principles of feature engineering: feature engineering vs. feature extraction; feature interaction — product, ratio, difference features; iterative feature construction process. □ Numerical features: standardisation, min-max, robust scaling; log, Box-Cox, Yeo-Johnson transformations; polynomial features; binning (equal-width, equal-frequency, decision tree-based); rolling and cumulative aggregations. □ Categorical features: one-hot encoding (OHE), ordinal encoding, target encoding (with smoothing formula and out-of-fold computation), frequency encoding, hashing trick, embedding encoding; dummy variable trap. □ Datetime and text features: datetime decomposition (year, month, day, hour, day-of-week, is_weekend); cyclical encoding (sin/cos); BoW, TF-IDF, n-grams, character n-grams; NER-derived features from text.

	Source: TB1: Chapters 2, 4 (Géron — feature scaling, encoding, sklearn transformers); TB4: Chapters 1–5 (Zheng & Casari — feature engineering for ML, text and categorical features)
AI Integration	Use ChatGPT to explain target encoding's leakage problem and out-of-fold computation; use Claude to design a complete feature engineering plan for a given prediction problem; use an AI assistant to explain Box-Cox vs. Yeo-Johnson and when each applies.

UNIT – IV	**FEATURE SELECTION AND DIMENSIONALITY REDUCTION**	**9 Hrs**

Content	□ Principles: curse of dimensionality; three families — filter (independent of model, fast), wrapper (model performance-based, accurate), embedded (selection inside training, balanced). Filter methods: variance threshold; chi-square (categorical features vs. categorical target); ANOVA F-test (numerical vs. categorical target); mutual information; Pearson/Spearman correlation — remove features with pairwise $\|r\| > 0.9$. □ Wrapper methods: forward selection, backward elimination, Recursive Feature Elimination (RFE), RFECV (RFE with cross-validation). Embedded methods: Lasso L1 (drives irrelevant coefficients to zero); Ridge L2 (shrinks but not zeros); Elastic Net; tree-based feature importance (MDI); permutation importance (unbiased, model-agnostic); SHAP values. □ Dimensionality reduction: PCA (linear, preserves variance, eliminates multicollinearity); LDA (supervised, maximises class separation); t-SNE (non-linear, local structure, visualisation only); UMAP (non-linear, local and global structure, learnable transform); autoencoders for non-linear reduction. **Source:** TB1: Chapter 4 (Géron — dimensionality reduction — PCA, LDA, t-SNE, autoencoders); TB3: Chapters 11–13 (Kuhn & Johnson — feature selection methods, filter, wrapper, embedded)
AI Integration	Use ChatGPT to explain SHAP values using Shapley game theory with a 3-feature model example; use Claude to compare PCA, LDA, t-SNE, and UMAP and derive a selection rule; use an AI assistant to explain why Lasso drives coefficients to zero but Ridge does not.

UNIT – V	**AUTOMATED FEATURE ENGINEERING AND RESPONSIBLE PRACTICES**	**9 Hrs**

Content	□ Automated feature engineering: Deep Feature Synthesis (DFS — Featuretools): entity sets, aggregation primitives (sum, mean, count across one-to-many relationships), transformation primitives (month, cumsum, diff); AutoFE tools — AutoFeat, TPOT, H2O AutoML. LLM-assisted feature ideation: prompting an LLM with dataset description and prediction task to generate candidate features — each must be validated for computability, leakage, domain validity, and empirical informativeness. □ Feature pipelines: scikit-learn Pipeline (atomic fit-transform chain — prevents leakage); ColumnTransformer (different transformers per column subset); feature leakage types: target leakage, temporal leakage, preprocessing leakage; feature stores: offline store (Parquet/warehouse), online store (Redis/DynamoDB), feature registry, point-in-time correct joins — Feast, Hopsworks. □ Responsible feature engineering: proxy features and disparate impact (80% rule); re-identification risk from feature combinations (Sweeney 2000 — 87% re-

	identification from 3 fields); feature documentation (definition, source, owner, known biases); GDPR Article 22; India DPDP Act 2023. **Source:** TB1: Chapter 2 (Géron — sklearn Pipeline and ColumnTransformer); TB5: Chapters 5, 11 (Huyen — feature stores, training-serving skew, responsible ML)
AI Integration	Use ChatGPT to explain DFS primitives for a customer→orders→products entity set; use Claude to evaluate AI-suggested features for leakage, proxy discrimination, and domain validity; use an AI assistant to design a GDPR and DPDP Act 2023 compliant feature engineering checklist.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1, CO5**	Define MCAR/MAR/MNAR; describe DFS primitives; explain GDPR Article 5.
L3	Apply	**CO2, CO3**	Apply MICE imputation; engineer cyclical features; implement TF-IDF vectorisation.
L4	Analyse	**CO2–CO4**	Analyse SMOTE vs. ADASYN; compare RFECV vs. Lasso; compare PCA vs. UMAP.
L5	Evaluate	**CO4**	Evaluate dimensionality reduction methods on MNIST using cluster separation.
L6	Create	**CO3, CO5**	Design a leakage-free feature engineering pipeline for a complete ML workflow.

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	L2	3	2	1	2	2	–	–	–	–	1	–
CO2	L4	3	3	2	2	3	–	–	–	1	1	–
CO3	L4	3	3	3	2	3	–	–	–	1	2	–
CO4	L5	3	3	3	3	3	–	–	–	1	2	–
CO5	L2	2	2	2	2	3	2	–	2	1	2	1

CO-PO Correlation Highlights	Strong PO5 (Modern Tools) through sklearn, Featuretools, SHAP, Fairlearn, and UMAP. PO2 (Problem Analysis) through feature selection comparison and leakage identification. PO6 (Engineer & Society) and PO8 (Ethics) through proxy feature audit, disparate impact measurement, and GDPR/DPDP Act 2023 governance checklist.

LEARNING RESOURCES

Textbooks

1. TB1: Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2023.
2. TB2: Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022.
3. TB3: Max Kuhn and Kjell Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, CRC Press, 2019. (Free at feat.engineering)
4. TB4: Alice Zheng and Amanda Casari, Feature Engineering for Machine Learning, O'Reilly Media, 2018.

Reference Books

1. TB5: Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022.
2. Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, 2nd Edition, Springer. (Free at hastie.su.domains/ElemStatLearn)

Online Resources

- Featuretools (DFS) – featuretools.com | SHAP – shap.readthedocs.io
- scikit-learn Feature Selection – scikit-learn.org/stable/modules/feature_selection
- imbalanced-learn (SMOTE, ADASYN) – imbalanced-learn.org |ydata-profiling – docs.profiling.ydata.ai
- Kaggle Learn: Feature Engineering (free) – kaggle.com/learn/feature-engineering
- NPTEL – Machine Learning, IIT Madras – nptel.ac.in | India DPDP Act 2023 – meity.gov.in

Course Code	INTELLIGENT DATABASE MANAGEMENT SYSTEMS	L	T	Pr	P	Credits
26AAI30402T	Common to CSE and Allied Branches	3	0	0	0	3

Course Objectives

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

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 DBMS architecture, data models, and ER modelling; design ER diagrams and map them to relational schemas. (L4)	L4 (Analyse)
CO2	Apply relational algebra and write SQL queries for data definition, manipulation, retrieval, and control. (L3)	L4 (Analyse)
CO3	Apply normalization techniques — 1NF through BCNF — to eliminate redundancy and design efficient relational schemas. (L4)	L4 (Analyse)
CO4	Analyse transaction schedules for serializability; apply concurrency control protocols and recovery mechanisms. (L4)	L4 (Analyse)
CO5	Evaluate modern database systems including NoSQL and vector databases; apply AI-assisted querying and data governance. (L5)	L5 (Evaluate)

Course Content

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.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define DBMS components; describe ER diagram elements and relational model.
L3	Apply	**CO2, CO3**	Write SQL queries; apply normalisation steps from UNF to BCNF.
L4	Analyse	**CO1–CO4**	Analyse conflict serializability; compare NoSQL models; examine query plans.
L5	Evaluate	**CO5**	Evaluate NoSQL vs. SQL for AI workloads; assess vector DB for semantic search.
L6	Create	**CO1, CO5**	Design a fully normalised schema with vector search integration for an AI application.

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's Lvl	PO1 Eng.	PO2 Problem	PO3 Design/	PO4 Invest-	PO5 Modern	PO6 Engineer	PO7 Environ.	PO8 Ethics	PO9 Team	PO10 Commun.	PO11 PrjMgm

		Know.	Analysis	Dev.	igation	Tools	& Society	& Sust.		Work		
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

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.

LEARNING RESOURCES

Textbooks

1. TB1: Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill, 2019.
2. TB2: 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

- NPTEL – Database Management Systems, IIT Madras – nptel.ac.in
- MySQL Documentation – dev.mysql.com/doc | MongoDB – docs.mongodb.com
- Stanford – Introduction to Databases – online.stanford.edu

India DPDP Act 2023 – meity.gov.in | FAISS – github.com/facebookresearch/faiss

Course Code	STREAMING DATA ENGINEERING AND PIPELINES	L	T	Pr	P	Credits
26AAI30403T	For AI& DS Branch	2	0	2	0	3

Prerequisite	Python Programming; Data Structures; Probability and Statistics; Databases

Course Objectives

- Introduce streaming data fundamentals — event streams, time semantics, windowing, and delivery guarantees — as distinct from batch processing.
- Develop competency in Apache Kafka for distributed event streaming: producers, consumers, topics, partitions, Kafka Streams, and Kafka Connect.
- Build knowledge of stream processing engines — Apache Spark Structured Streaming and Apache Flink — for stateful, fault-tolerant, real-time computation.
- Apply streaming pipelines to AI/ML workflows: real-time feature engineering, online inference, and streaming feature stores.
- Develop understanding of pipeline observability, schema evolution, data quality, and responsible data practices in streaming systems.

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 streaming data fundamentals — event time, watermarks, windowing, and exactly-once semantics — and contrast with batch processing. (L2)	L2 (Understand)
CO2	Design and configure Apache Kafka topics, producers, consumers, and connectors for distributed event streaming applications. (L4)	L2 (Understand)
CO3	Apply Spark Structured Streaming and Apache Flink for stateful stream processing, windowed aggregations, and stream joins. (L3)	L4 (Analyse)
CO4	Design real-time AI/ML pipelines integrating streaming feature engineering, online inference, and streaming feature stores. (L4)	L4 (Analyse)
CO5	Evaluate streaming pipeline reliability, schema evolution, data quality, and regulatory compliance for production deployment. (L5)	L5 (Evaluate)

Course Content

UNIT – I	STREAMING DATA FUNDAMENTALS	9 Hrs

Content	
	- Batch vs. stream processing: latency, throughput, bounded vs. unbounded data; use cases — fraud detection, IoT telemetry, real-time recommendations. - Event time vs. processing time vs. ingestion time; out-of-order events; watermarks; late data handling strategies. - Windowing: tumbling (fixed, non-overlapping), sliding (overlapping), session (gap-based); window triggering and allowed lateness. - Delivery semantics: at-most-once, at-least-once, exactly-once; idempotent

	producers; transactional consumers; exactly-once challenges in distributed systems.
AI Integration	Use ChatGPT to explain event time vs. processing time with an IoT example; use Claude to illustrate windowing types with timeline diagrams; use an AI assistant to explain exactly-once semantics and distributed system challenges.

UNIT – II	**APACHE KAFKA — DISTRIBUTED EVENT STREAMING**	**9 Hrs**

Content	• Kafka architecture: brokers, KRaft (Kafka 3.x+), topics, partitions, replication, leader/follower replicas, log-based storage. • Producers: serialization, partitioner strategies, acks (0/1/all), idempotent producer. Consumers: consumer groups, partition assignment, offset management, auto vs. manual commit. • Kafka Streams DSL: KStream, KTable, GlobalKTable; stateful aggregations and joins; changelog topics; state stores; interactive queries. • Kafka Connect: source/sink connectors; Debezium CDC, JDBC source, S3/Elasticsearch sinks; Schema Registry (Avro/Protobuf); MirrorMaker 2 for multi-cluster replication.
AI Integration	Use ChatGPT to explain consumer group rebalancing; use Claude to design a Kafka topic schema for a ride-hailing event stream; use an AI assistant to explain CDC with Debezium for real-time database change capture.

UNIT – III	**STREAM PROCESSING ENGINES**	**9 Hrs**

Content	• Spark Structured Streaming: unified batch-stream API; unbounded DataFrames; trigger modes (micro-batch, continuous); output modes (append, update, complete); checkpointing. • Spark stateful operations: mapGroupsWithState, flatMapGroupsWithState; windowed aggregations; watermarks; stream-stream and stream-static joins. • Apache Flink: true streaming (record-at-a-time); job graph (sources, transformations, sinks); event time, watermarks, allowed lateness; Flink SQL and Table API. • Flink state management: keyed state (ValueState, ListState, MapState); operator state; incremental checkpointing; savepoints for versioned state and pipeline upgrades.
AI Integration	Use ChatGPT to compare Spark Structured Streaming vs. Flink for low-latency fraud detection; use Claude to explain Flink checkpointing and savepoints; use an AI assistant to design a stateful windowed aggregation for IoT sensor data.

UNIT – IV	**STREAMING PIPELINES FOR AI AND ML**	**9 Hrs**

Content	□ Real-time feature engineering: rolling statistics, sessionization, aggregations over time windows; consistency challenges between streaming and batch features. □ Streaming feature stores: dual-write architecture; online store (Redis, DynamoDB) for inference; offline store (Parquet/warehouse) for training; point-in-time correct retrieval; Feast and Hopsworks streaming ingestion. □ Real-time inference pipelines: REST API inference (TorchServe, Triton); event-driven inference (Kafka → model → output topic); feature fetching from online store at inference time. □ Online learning and model updating: concept drift detection (DDM, ADWIN, Page-Hinkley); incremental model updates; River library for online ML; retraining triggers.
AI Integration	Use ChatGPT to explain the dual-write pattern in streaming feature stores; use Claude to design a real-time fraud detection pipeline from Kafka through feature store to model server; use an AI assistant to explain ADWIN concept drift detection.

UNIT – V	**RELIABILITY, OBSERVABILITY, AND RESPONSIBLE STREAMING**	**9 Hrs**

Content	□ Pipeline observability: throughput, lag, latency, error rate metrics; structured logging; distributed tracing; Prometheus + Grafana dashboards; consumer lag monitoring. □ Schema evolution: backward, forward, and full compatibility in Avro/Protobuf; Confluent Schema Registry enforcement; breaking vs. non-breaking changes. □ Data quality in streams: late data, duplicates, nulls, schema violations, data skew; Great Expectations for streaming validation; dead-letter queues (DLQ); circuit breaker patterns. □ Responsible streaming: GDPR and India DPDP Act 2023 on real-time personal data — consent, purpose limitation, right to erasure in event streams; PII masking and tokenization; TLS encryption in transit.
AI Integration	Use ChatGPT to design a Prometheus + Grafana setup for Kafka pipeline monitoring; use Claude to explain schema compatibility modes with Avro examples; use an AI assistant to design a GDPR-compliant pipeline with PII masking and erasure support.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1, CO2**	Define event streaming, Kafka topics, partitions, consumer groups, Flink state.
L3	Apply	**CO3, CO4**	Implement Kafka producer-consumer; configure Spark Structured Streaming with watermarks.
L4	Analyse	**CO3, CO4**	Compare Kafka vs. Pulsar; analyse Flink stateful operator checkpointing.
L5	Evaluate	**CO5**	Evaluate monitoring dashboards; assess GDPR erasure

			strategy for streaming systems.
L6	Create	**CO4, CO5**	Design a complete AI feature streaming pipeline with observability and 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	L2	3	2	1	2	2	–	–	–	–	1	–
CO2	L2	3	2	1	2	2	–	–	–	–	1	–
CO3	L4	3	3	3	2	3	1	–	–	1	2	1
CO4	L4	3	3	3	3	3	1	–	–	1	2	1
CO5	L5	3	3	3	3	3	2	1	2	1	2	1

CO-PO Correlation Highlights	Strong PO5 through Kafka, Apache Flink, Spark Structured Streaming, Prometheus, Grafana, and Evidently AI. PO3 (Design/Development) through streaming pipeline architecture and ML feature pipeline design. PO6 and PO8 (Ethics) through GDPR right-to-erasure, data lineage, and responsible streaming governance in Unit V.

LEARNING RESOURCES

Textbooks

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017.
2. Gwen Shapira et al., Kafka: The Definitive Guide, 2nd Edition, O'Reilly Media, 2021.
3. Fabian Hueske and Vasiliki Kalavri, Stream Processing with Apache Flink, O'Reilly Media, 2019.

Reference Books

1. Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022.
2. Bill Chambers and Matei Zaharia, Spark: The Definitive Guide, O'Reilly Media, 2018.
3. Tyler Akidau, Slava Chernyak, Reuven Lax, Streaming Systems, O'Reilly Media, 2018.

Online Resources

- Apache Kafka – kafka.apache.org | Apache Flink – nightlies.apache.org/flink/flink-docs-stable
- Spark Structured Streaming – spark.apache.org/docs/latest/structured-streaming-programming-guide.html
- Confluent Developer – developer.confluent.io | Feast – docs.feast.dev | River – riverml.xyz
- India DPDP Act 2023 – meity.gov.in | GDPR – gdpr.eu

Course Code	FOUNDATIONS OF ARTIFICIAL INTELLIGENCE	L	T	Pr	P	Credits
26AAI30404T	For AI& DS, CSE(AI), CSE (DS), CSE (AI&ML), AI & ML,& AI Branches	2	0	2	0	3

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 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)

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.
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.
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.
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.
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.
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.

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)

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	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

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. Ian 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

Course Code	INTELLIGENT DATA PREPARATION AND FEATURE ENGINEERING — LAB	L	T	Pr	P	C
26AAI30401P	For AI& DS Branch	0	0	0	3	1.5

Lab Course Outcomes

- Conduct EDA and produce an AI-readiness profile using automated profiling tools on a real dataset.
- Apply missing data classification, outlier detection, class imbalance handling, and data integration techniques.
- Engineer numerical, categorical, datetime, and text features; validate leakage prevention using sklearn Pipeline.
- Apply filter, wrapper, and embedded feature selection; compare PCA, LDA, t-SNE, and UMAP projections.
- Build an automated feature engineering pipeline using Featuretools DFS; produce a responsible FE governance checklist.

List of Experiments

Unit I Data Collection, Exploration, and Profiling

1. **Automated EDA and Data Quality Profiling**
   Load the Titanic dataset; generate aydata-profiling HTML report and identify the top 5 quality issues. Implement a custom audit function reporting all six quality dimensions (accuracy, completeness, consistency, validity, uniqueness, timeliness). Classify each missing column as MCAR, MAR, or MNAR using correlation between the missingness indicator and other features. Produce an AI-readiness scorecard: class balance ratio, missing rate per feature, highly correlated pairs ($|r| > 0.7$).

2. **Univariate, Bivariate, and Multivariate EDA**
   Using the Ames Housing dataset, compute skewness and kurtosis for all numerical columns and plot histograms with KDE. Compute Pearson, Spearman, and Kendall correlations between SalePrice and all numerical features; plot scatter plots for the top 5. Generate a pair plot for 6 key features and a correlation heatmap; identify multicollinear pairs ($|r| > 0.85$). If a datetime column is present, decompose it into trend, seasonality, and residual and plot ACF and PACF.

3. **Data Collection via REST API and Bias Audit**
   Access the Open-Meteo weather API to fetch 30 days of hourly temperature and humidity for two cities; compute daily aggregates and plot a comparison line chart. Load a public dataset from data.gov.in; report shape, dtypes, null rates, and value ranges. Identify which of the four collection biases (selection, survivorship, response, temporal) apply to the weather data; document 3 specific risks and propose a mitigation for each.

Unit II Data Cleaning, Preprocessing, and Integration

4. **Missing Data Classification and Imputation Strategies**
   On the Titanic dataset, visualise missing data using a missingno matrix. Compare 5 imputation strategies for Age — mean, median, mode, KNN (k=5), and MICE — by reporting post-imputation distribution shape (KS test). Add a binary missingness indicator column before imputing; compare validation AUC (LogisticRegression) with and without the indicator. Justify treating the Cabin column (77% missing) as MNAR and document the deletion decision.

5. **Outlier Detection, Noise Handling, and Class Imbalance**

Detect outliers in GrLivArea using Z-score, IQR, and Isolation Forest; compare counts and overlap. Apply Winsorisation vs. log transformation; compare skewness before and after. On a credit card fraud dataset, train LogisticRegression without resampling, then with SMOTE, ADASYN, and Tomek links separately; compare precision, recall, F1, PR-AUC, and ROC-AUC. Plot PR curves for all approaches on the same axes.

6. Data Integration, Normalisation, and Discretisation

Merge customers, orders, and product CSV files; verify referential integrity and detect duplicate customers using exact and fuzzy matching (Levenshtein distance). Create 3 interaction features: TotalOrderValue, CustomerLifetimeValue, and DaysSinceLastOrder. Compare StandardScaler, MinMaxScaler, and RobustScaler on 4 outlier-containing columns by training KNN and reporting accuracy. Apply equal-width, equal-frequency, and decision tree-based binning to Age; compare information gain.

Unit III | Feature Engineering for Structured Data

7. Numerical Feature Engineering

On Ames Housing, identify columns with |skew| > 0.75; apply log1p, Box-Cox, and Yeo-Johnson; report post-transformation skewness and choose the best per column. Apply PolynomialFeatures(degree=2) on 5 features; compare LinearRegression R^2 before and after with Ridge regularisation. Create interaction features (PricePerSqFt, TotalBath, HouseAge); compare LinearRegression R^2 with and without them.

8. Categorical and Datetime Feature Engineering

OHE Neighborhood; apply OrdinalEncoder to ExterQual and verify coefficient sign. Apply out-of-fold target encoding to Neighborhood with smoothing (k=10); compare R^2 against OHE to verify leakage prevention. On a datetime dataset, extract year, month, hour, day_of_week, is_weekend; apply cyclical sin/cos encoding to hour and day; compare RMSE of LinearRegression on integer vs. cyclical hour encoding.

9. Text Feature Engineering and Feature Interactions

On the 20 Newsgroups dataset, apply CountVectorizer (BoW) and TfidfVectorizer (max_features=5000); train Multinomial Naive Bayes; compare precision, recall, and F1. Apply unigram, bigram (1,2), and trigram (1,3) TF-IDF; report which n-gram range is optimal. On Titanic, engineer FamilySize, IsAlone, Title (regex), AgeGroup, and FarePerPerson; compare LogisticRegression AUC with and without interaction features.

Unit IV | Feature Selection and Dimensionality Reduction

10. Filter Methods and Correlation-Based Selection

On Breast Cancer Wisconsin (30 features), apply VarianceThreshold; rank features by chi-square, ANOVA F-test, and mutual information. Compute Jaccard similarity between the three top-10 lists. Remove one feature from each correlated pair (|r| > 0.90). Compare LogisticRegression (5-fold CV) accuracy on all 30 features, top-10 by mutual information, and the correlation-filtered set; plot accuracy vs. number of features.

11. Wrapper and Embedded Methods with SHAP

On Pima Indians Diabetes (8 features), apply RFECV; plot validation accuracy vs. feature count. Fit Lasso at alpha = 0.001, 0.01, 0.1, 1.0; plot coefficient path; compare Lasso-selected features with RFECV. Train RandomForestClassifier; compare MDI importance with permutation importance (100 repeats). Compute SHAP values using TreeExplainer; generate summary plot, bar plot, and dependence plot for the top feature.

12. Dimensionality Reduction — PCA, LDA, t-SNE, and UMAP

On a 5,000-sample MNIST subset (784 features), apply PCA retaining 95% variance; plot cumulative explained variance. Apply LDA (max 9 components); compare 2D scatter plots for PCA and LDA

coloured by digit class. Apply t-SNE (perplexity=30) and UMAP (n_neighbors=15, min_dist=0.1); compare 2D projections on visual cluster separation and computation time. Train LogisticRegression on PCA and LDA components; compare accuracy with the full 784-feature baseline.

Unit V	Automated Feature Engineering and Responsible Practices

13. Deep Feature Synthesis with Featuretools

Build a FeaturetoolsEntitySet from customers, orders, and order_items tables. Run DFS (depth=2) with SUM, MEAN, COUNT, MAX, MIN aggregation primitives and MONTH, DIFF, YEAR transformation primitives; report total features generated. Identify 5 generated features that could cause temporal leakage if target is customer churn within 30 days. Filter zero-variance, >50% null, and high-correlation features; compare LogisticRegression AUC on DFS features vs. manually engineered features.

14. End-to-End Feature Pipeline with Leakage Demonstration

Build a correct sklearn Pipeline on Titanic: SimpleImputer → OHE → StandardScaler → LogisticRegression using ColumnTransformer; evaluate with 5-fold CV and report mean AUC. Deliberately introduce preprocessing leakage (scale on full dataset before split); measure AUC inflation. Introduce target leakage (add a feature derived from the target); measure AUC jump. Document the magnitude of inflation caused by each leakage type.

15. Responsible Feature Engineering — Bias Detection and Governance

On the UCI Adult Income dataset, compute correlation of every feature with sex; identify proxy features ($|r| > 0.3$). Train LogisticRegression without sex and without proxy features; compute disparate impact ratio (selection rate female/male); check against the 80% rule using FairlearnMetricFrame. Demonstrate re-identification risk: combine age, education, occupation into a composite key; report the fraction of uniquely identifiable records. Produce a feature documentation card (definition, bias risk, proxy risk, GDPR Article 5 note, India DPDP Act 2023 obligation) as a Pandas DataFrame saved to CSV.

AI Tools Integration (Lab)	Use Google Colab as the primary lab environment. Use ChatGPT or Claude to explain missing data classification, interpret SHAP plots, and debug Pipeline errors. Use Claude to evaluate AI-suggested features for leakage, proxy discrimination, and domain validity in Experiment 13. Use an AI assistant to design a GDPR and India DPDP Act 2023 compliance checklist in Experiment 15 and verify it against the legal text. All AI-generated code must be tested and documented before submission.

LEARNING RESOURCES

Textbooks

1. TB1: Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2023.
2. TB2: Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022.
3. TB3: Max Kuhn and Kjell Johnson, Feature Engineering and Selection, CRC Press, 2019. (Free at feat.engineering)
4. TB4: Alice Zheng and Amanda Casari, Feature Engineering for Machine Learning, O'Reilly Media, 2018.

Reference Books

1. Chip Huyen, Designing Machine Learning Systems, O'Reilly Media, 2022.

2. Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, 2nd Edition, Springer. (Free at hastie.su.domains/ElemStatLearn)

Online Resources

- Google Colab – colab.research.google.com |ydata-profiling – docs.profiling.ydata.ai
- imbalanced-learn – imbalanced-learn.org |Featuretools – featuretools.com | SHAP – shap.readthedocs.io
- Fairlearn – fairlearn.org | UMAP – umap-learn.readthedocs.io | NPTEL – nptel.ac.in
- Kaggle Learn: Feature Engineering – kaggle.com/learn/feature-engineering
- India DPDP Act 2023 – meity.gov.in | GDPR – gdpr.eu

Course Code	INTELLIGENT DATABASE MANAGEMENT SYSTEMS LAB	L	T	Pr	P	Credits
26AAI30402P	Common for CSE & Allied Branches	0	0	0	3	1.5

- **Prerequisites:** Data Structures.
- **Branch:** Common to CSE and Allied Branches | **Regulation:** JNTUA R26

Course Objectives

- Develop practical skills in database design and implementation through ER modelling, relational schema design, and database creation for real-world applications.
- Apply SQL techniques for database operations including data definition, manipulation, querying, optimization, indexing, and transaction control.
- Design efficient and reliable database systems using normalization, schema decomposition, concurrency control, and recovery mechanisms.
- Explore modern intelligent database technologies including NoSQL databases, graph databases, vector databases, and AI-enabled data management systems.
- 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:

- **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.
- **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.
- **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:

- **Experiment 4: Joins & Subqueries —** Develop SQL queries using joins, nested queries, subqueries, set operations, aggregate functions, GROUP BY, and HAVING clauses.
- **Experiment 5: Views, Indexes & Optimization —** Create and analyze views and indexes, and perform query optimization comparing indexed and non-indexed access methods.
- **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:

- **Experiment 7: Normalization (1NF–BCNF) —** Identify functional dependencies and perform normalization of relations from 1NF through BCNF.
- **Experiment 8: Closures & Decomposition —** Implement attribute closure computation, minimal cover generation, lossless decomposition, and dependency preservation.
- **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:

- **Experiment 10: Transaction Control —** Implement transaction control operations using COMMIT, ROLLBACK, and SAVEPOINT.
- **Experiment 11: Concurrency Control —** Simulate concurrency control mechanisms including locking protocols, two-phase locking, deadlock detection, and serializability analysis.
- **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:

- **Experiment 13: NoSQL with MongoDB —** Perform CRUD operations and aggregation queries using a NoSQL database such as MongoDB.
- **Experiment 14: Graph Databases with Neo4j —** Develop a graph database application using Neo4j and implement Cypher queries.
- **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:

- An AI-assisted Text-to-SQL query interface for a chosen database schema.
- 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	PSO1	PSO2
CO1	L3	3	2	3	2	3	–	–	–	1	1	–	2	3	2
CO2	L3	3	2	3	2	3	–	–	–	1	1	–	2	3	2
CO3	L3	3	3	2	2	3	–	–	–	1	1	–	2	3	2
CO4	L4	3	3	3	2	3	–	–	–	1	1	–	2	3	2
CO5	L4	3	3	3	3	3	–	–	1	2	2	1	3	3	3

Textbooks

- Abraham Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw-Hill.
- Ramez Elmasri and Shamkant Navathe, Fundamentals of Database Systems, 7th Edition, Pearson.

Reference Books

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

Online Resources

- NPTEL – Database Management System – nptel.ac.in
- MongoDB University – Free NoSQL Courses – university.mongodb.com
- Neo4j GraphAcademy – Free Graph Database Courses – graphacademy.neo4j.com

Recommended Free & Open-Source AI / Computational Tools

- **AI Assistants:** ChatGPT; GitHub Copilot – query generation, debugging, and schema/normalization validation.
- **Modelling Tools:** draw.io – ER diagram design.
- **Databases & Engines:** MongoDB (Atlas free tier); Neo4j (Community/AuraDB Free) – NoSQL and graph database experimentation.
- **Vector Search Libraries:** FAISS; Chroma – similarity search and vector database applications.
- **Frameworks:** LangChain – building Text-to-SQL and RAG-over-SQL pipelines.
- **Notebooks & Simulation:** Python & Jupyter Notebook – transaction simulations and normalization/closure computations.

DATA ANALYTICS USING PYTHON

Course Code	Year/Sem	L	T	Pr	P	Credits
26ACS05405S	—	1	0	0	2	2

Prerequisites: Basic knowledge of Python programming and programming fundamentals. (Inferred – not specified in the source.)
Branch: Common to CSE & Allied Branches | Regulation: JNTUA R26

Course Objectives

- Develop practical skills for handling, processing, and analyzing data using Python-based analytical tools and libraries.
- Enable students to perform data cleaning, transformation, aggregation, and exploratory analysis on structured datasets.
- Build competency in creating visualizations and dashboards for effective interpretation and communication of analytical insights.
- Introduce real-world analytics workflows involving data acquisition, processing, reporting, and presentation.
- 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:

- **Experiment 1: Dataset Import and Exploration —** Import datasets from CSV, Excel and JSON formats. Examine dimensions, data types, missing values and generate summary statistics.
- **Experiment 2: Numerical Data Processing using NumPy —** Perform array creation, indexing, slicing, aggregation, mathematical operations and vectorized computations.
- **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:

- **Experiment 4: Data Cleaning and Missing Value Handling —** Identify duplicate records, missing values and outliers. Apply cleaning and preprocessing techniques.
- **Experiment 5: Data Transformation and Feature Construction —** Perform grouping, merging, aggregation, reshaping and derived column generation.
- **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:

- **Experiment 7: Data Visualization using Matplotlib —** Generate visual reports using line charts, bar charts, histograms and pie charts.
- **Experiment 8: Advanced Visualization using Seaborn —** Develop scatter plots, boxplots, heatmaps and distribution analysis.
- **Experiment 9: Interactive Visualization and Dashboard Design —** Build interactive visualizations and analytical dashboards.
- **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:

- **Experiment 11: Data Collection using APIs and JSON Processing —** Collect data using APIs and process JSON data.
- **Experiment 12: Time-Series and Trend Analysis —** Analyze temporal datasets and generate trend reports including date handling, rolling statistics and visualization.
- **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:

- **Experiment 14: Dashboard Development using Real-world Datasets —** Create complete dashboards for decision support using Education, Healthcare, Retail or Weather datasets.
- **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	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	L4	3	2	2	2	3	–	–	–	1	1	–	2	3	2
CO2	L5	3	3	2	2	3	–	–	–	1	1	–	2	3	2
CO3	L5	3	3	2	3	3	–	–	–	1	2	1	2	3	2
CO4	L6	3	3	3	3	3	–	–	–	1	3	2	3	3	3
CO5	L6	3	3	3	3	3	1	–	1	2	3	3	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

- Python Documentation – docs.python.org
- Jupyter Documentation – docs.jupyter.org
- NumPy Documentation – numpy.org/doc

- Pandas Documentation – pandas.pydata.org/docs
- Matplotlib Documentation – matplotlib.org/stable
- Seaborn Documentation – seaborn.pydata.org
- Plotly Documentation – plotly.com/python

Recommended Free & Open-Source AI / Computational Tools

Core Libraries: Python, Pandas, NumPy – data handling, manipulation and numerical computing.

Visualization: Matplotlib, Seaborn, Plotly – static and interactive charts and dashboards.

Environment: Jupyter Notebook – interactive analysis, experimentation and reporting.

Data Access: Requests – API-based data collection and JSON processing.

AI Assistants: ChatGPT, Claude, GitHub Copilot – code generation, automatic data profiling and analytical assistance.

Course Code	DATA ANALYTICS USING PYTHON	L	T	Pr	P	Credits
26ACS05405S	Common to CSE Allied Branches	1	0	0	2	2

Course Objectives

- Develop practical skills for handling, processing, and analyzing data using Python-based analytical tools and libraries.
- Enable students to perform data cleaning, transformation, aggregation, and exploratory analysis on structured datasets.
- Build competency in creating visualizations and dashboards for effective interpretation and communication of analytical insights.
- Introduce real-world analytics workflows involving data acquisition, processing, reporting, and presentation.
- Prepare students with a strong foundation in data analytics for advanced courses in Artificial Intelligence, Machine Learning, and Data Science.

Course Outcomes (COs

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Import, process, and manipulate datasets using Python analytical libraries. (L4)	L4 (Analyse)
CO2	Apply data cleaning, transformation, and aggregation techniques to prepare datasets for analysis. (L5)	L5 (Evaluate)
CO3	Perform exploratory data analysis and create meaningful visualizations to derive insights. (L5)	L5 (Evaluate)
CO4	Develop analytical reports and dashboards using Python tools and AI-assisted analytics workflows. (L6)	L6 (Create)
CO5	Design and implement data-driven analytical solutions using real-world datasets. (L6)	L6 (Create)

AI Tools Integration (Lab)	Dataset understanding and profiling: ChatGPT and Claude — explain column meanings, suggest data quality checks, recommend imputation strategies. Visualisation recommendation: ChatGPT and Perplexity AI — suggest appropriate chart types for given analysis questions; verify recommendations empirically. Analytical assistance: NotebookLM — upload datasets, documentation, or reports as sources; generate study guides, audio overviews, and grounded Q&A without hallucination. Code generation: GitHub Copilot — autocomplete for Pandas, Matplotlib, Seaborn, and Plotly code; always test generated code before submission. Report generation: Claude or ChatGPT — generate executive summaries and insight narratives from data descriptions; verify all statistical claims independently. Responsible use: all AI-generated analysis must be verified against actual data; do not report AI-suggested insights without computing the supporting statistics yourself. An AI usage log is required for Experiments 13 and 15.

Course Content

Module I – Data Handling Using Python (Sessions 1–3)	
Tools	Python 3, Jupyter Notebook, Pandas, NumPy
Topics	Data Analytics Workflow; Jupyter Notebook Environment; Data Sources and File Formats; Importing Data; CSV, Excel and JSON Processing; NumPy Arrays; Data Manipulation; Vectorised Operations; Data Export.

1. **Dataset Import and Exploration**
   Import datasets from CSV (pd.read_csv), Excel (pd.read_excel), and JSON (pd.read_json) formats into Pandas DataFrames. Examine dataset dimensions (shape), column names (columns), data types (dtypes), and memory usage (info()). Compute summary statistics (describe()) for numerical columns; count missing values per column (isnull().sum()); display the first and last 5 rows (head(), tail()). Export the cleaned dataset to a new CSV and JSON file using to_csv() and to_json(). Use ChatGPT or NotebookLM to explain any unfamiliar statistics in the describe() output; document each explanation.

2. **Numerical Data Processing using NumPy**
   Create 1D and 2D NumPy arrays from Python lists and using np.arange(), np.linspace(), np.zeros(), np.ones(), np.eye(). Perform indexing (arr[i]), slicing (arr[1:4]), and fancy indexing (arr[[0,2,4]]). Apply aggregation functions: np.sum(), np.mean(), np.std(), np.min(), np.max(), np.median() on both axes. Implement vectorised operations: element-wise arithmetic, broadcasting a scalar across an array, and computing the dot product of two 2D arrays. Compare execution time of a NumPy vectorised operation vs. an equivalent Python loop for array of 1 million elements using time module.

3. **DataFrame Operations using Pandas**
   Create a DataFrame from a dictionary with 6 columns and 20 rows of student data (roll_no, name, branch, marks_list, attendance, grade). Perform: column selection (single and multiple), row filtering using boolean conditions (marks > 60 and attendance >= 75), loc[] and iloc[] indexing, and sorting by marks descending. Compute group-level aggregations using groupby('branch').agg({'marks': ['mean','max','min'], 'attendance': 'mean'}); display results in a formatted table. Add a derived column pass_fail (Pass if marks >= 50, Fail otherwise) using np.where(); export the final DataFrame to Excel. Use Claude or ChatGPT to explain the difference between loc[] and iloc[]; verify the explanation using 3 examples.

Module II – Data Processing and Transformation using Pandas (Sessions 4–6)	
Tools	Pandas, Jupyter Notebook
Topics	Series and DataFrames; Data Selection; Indexing and Filtering; Missing Data Handling; Data Cleaning; Aggregation; Grouping; Merging; Sorting; Feature Construction.

4. **Data Cleaning and Missing Value Handling**
   Load the Titanic dataset (or any dataset with missing values and duplicates). Identify: duplicate rows (duplicated().sum()); missing values per column (isnull().sum(), visualised using a missingno bar chart); outliers in the fare/age column using Z-score (|z| > 3) and IQR (1.5×IQR) methods. Remove duplicate rows; impute missing Age with median, Embarked with mode; drop the Cabin column (>70% missing) and justify the decision. Apply Winsorisation to the Fare column (clip at 1st and 99th percentile); compare distribution before and after using histograms. Use ChatGPT to recommend the appropriate missing data strategy for each column; evaluate the suggestion and document whether you accepted or modified it.

5. **Data Transformation and Feature Construction**
   Using the Ames Housing or NYC Taxi dataset: apply melt() to convert a wide-format DataFrame to long format; verify row count; apply pivot_table() to reconstruct the wide format and compare. Merge two related DataFrames (e.g., customers and orders) using inner, left, right, and outer joins — report row

counts for each join type. Create 3 derived columns: price_per_sqft (price/area), age_of_house (current_year − year_built), is_recent (age_of_house ≤ 10). Apply groupby with transform() to add group-mean price as a new column without reducing the DataFrame. Use an AI assistant to suggest 2 additional feature ideas; implement both and validate that they are non-leaking (not derived from the target variable).

6. Descriptive Statistics and Data Analysis

Load any domain dataset (Iris, Wine Quality, or Student Performance). Compute: measures of central tendency — mean, median, mode using Pandas and scipy.stats; measures of dispersion — variance, standard deviation, range, IQR; shape statistics — skewness and kurtosis using df.skew() and df.kurtosis(). Generate a cross-tabulation of two categorical variables using pd.crosstab() with normalize=True; interpret the conditional frequency distribution. Identify columns with |skewness| > 1; apply log1p transformation to reduce skewness; compare before and after using describe(). Use NotebookLM — upload the dataset documentation as a source and ask it to explain each statistical measure in context; document the explanation.

Module III – Exploratory Data Analysis and Visualization (Sessions 7–10)	
Tools	Matplotlib, Seaborn, Plotly
Topics	Descriptive Statistics; Data Exploration; Correlation Analysis; Visualization Principles; Line Charts; Bar Charts; Scatter Plots; Histograms; Boxplots; Heatmaps; Interactive Visualization.

7. Data Visualization using Matplotlib

Using a sales or student dataset, create 4 chart types with Matplotlib: (a) line chart — monthly revenue trend with markers and a 3-month rolling average overlay; (b) bar chart — top 10 products by sales volume (horizontal, colour-coded by category); (c) histogram — distribution of marks with 20 bins and a KDE curve overlaid using scipy.stats.gaussian_kde; (d) pie chart — market share by product category. Apply chart design principles to each: proper axis labels, title, legend, colour palette (avoid red-green for accessibility), and grid lines. Use ChatGPT to recommend a chart type for 3 new analysis questions; implement the recommended charts and document whether each recommendation was appropriate.

8. Advanced Visualization using Seaborn

Using the Titanic or Tips dataset, create 5 Seaborn plots: (a) scatter plot with regression line (sns.regplot) — age vs. fare, coloured by survived; (b) boxplot (sns.boxplot) — fare distribution per passenger class; (c) violin plot (sns.violinplot) — age distribution per sex and survived; (d) heatmap (sns.heatmap) — correlation matrix of all numerical columns with annotations and a diverging colour palette; (e) pair plot (sns.pairplot) — all numerical columns coloured by survival status. For each plot, write a 2-sentence interpretation of the insight shown. Use an AI assistant to generate an alternate seaborn palette recommendation and test it on 2 charts.

9. Interactive Visualization and Dashboard Design

Using Plotly Express and Plotly Graph Objects, build 3 interactive charts on a real dataset: (a) interactive line chart (px.line) with hover tooltips showing exact values and a range slider for the x-axis; (b) interactive scatter plot (px.scatter) with size mapped to a third variable, colour mapped to a category, and a dropdown filter for year; (c) animated bar chart (px.bar with animation_frame) showing top-5 values changing over time. Combine all 3 charts into a Plotly subplot (make_subplots) dashboard. Export the dashboard as an HTML file viewable in any browser. Use ChatGPT to suggest the best Plotly chart type for 2 additional analysis questions; justify your implementation choice.

10. Correlation and Comparative Data Analysis

Using a multivariate dataset (Wine Quality, Boston Housing, or Student Performance): compute pairwise Pearson, Spearman, and Kendall correlation coefficients for all numerical columns; display as three separate annotated heatmaps and identify the top-3 and bottom-3 correlated feature pairs. Perform comparative analysis: compare mean marks across 3 student groups using a grouped bar chart with error

bars (±1 std); apply a t-test (scipy.stats.ttest_ind) to verify if the difference between two groups is statistically significant ($p < 0.05$). Identify 3 pairs of variables with strong correlation ($|r| > 0.7$); create scatter plots with regression lines for each. Document whether correlation implies causation for each pair; use Claude to verify your reasoning.

Module IV – Data Analytics Using Real-World Datasets (Sessions 11–13)	
Tools	Pandas, Requests, Plotly
Topics	Data Collection Methods; Public Dataset Usage; JSON Processing; API-based Data Collection; Time-Series Data Handling; Data Transformation; Report Generation; Dashboard Concepts.

11. Data Collection using APIs and JSON Processing

Collect data from two public REST APIs using the requests library: (a) Open-Meteo API (free, no key) — fetch 30 days of hourly temperature, humidity, and wind speed for two cities; load JSON response into a DataFrame using pd.json_normalize(); compute daily aggregates (min, max, mean); plot a dual-city comparison line chart; (b) REST Countries API (free, no key) — fetch all countries; extract name, capital, population, area, and region; load into a DataFrame; find the top 10 most populous countries per region. Handle API errors (timeout, 404, invalid city) with try-except; document all error cases and the handling applied.

12. Time-Series and Trend Analysis

Load a time-series dataset (daily stock prices from Yahoo Finance using yfinance, or provided weather/sales CSV). Parse the date column using pd.to_datetime() and set as DatetimeIndex. Perform: resampling — daily data to weekly mean and monthly sum using resample(); rolling statistics — 7-day and 30-day rolling mean using rolling().mean(); EWM — exponential weighted mean (ewm(span=7).mean()) to smooth short-term fluctuations. Decompose the time series into trend, seasonality, and residual using statsmodels.tsa.seasonal_decompose(). Plot all components on a 4-panel Matplotlib figure. Identify and annotate 3 significant events (peaks/troughs) on the trend line. Use ChatGPT to interpret the seasonality component; document the explanation.

13. Analytical Report Generation

Using the complete dataset from any previous experiment, generate a structured analytical report as a Jupyter Notebook: (a) Executive Summary — 5-sentence description of dataset, analysis goals, and key findings; (b) Dataset Overview — shape, column descriptions, data types, and quality summary (null rates, duplicate count, outlier count); (c) Key Metrics — top-5 summary statistics, distribution of target variable; (d) Insights — 5 numbered insights each supported by one chart; (e) Recommendations — 3 data-driven recommendations for a business decision-maker. Export the notebook as a PDF using nbconvert. Use NotebookLM — upload the notebook as a source; generate an audio overview; share the audio summary link and include the transcript as an appendix to the report.

Module V – Applied Data Analytics Project (Sessions 14–15)	
Tools	Python, Jupyter Notebook, Plotly
Topics	Analytics Workflow; Data Preparation; Data Visualization; Insight Generation; Dashboard Development; Documentation; Project Presentation.

14. Dashboard Development using Real-World Datasets

Select a domain dataset (Education — student performance, Healthcare — patient records, Retail — sales data, or Weather — daily weather data). Build a complete multi-panel Plotly dashboard (minimum 6 charts) covering: dataset overview (bar/pie chart of category distribution), trend analysis (line chart with rolling average), comparative analysis (grouped bar with error bars), correlation analysis (annotated heatmap), outlier identification (boxplot), and a KPI summary panel (4 metric cards showing totals,

averages, max/min). Add a dropdown filter (by category, year, or region) that updates all charts simultaneously. Export as an HTML file and a PDF report. Use an AI tool to suggest 2 additional KPIs relevant to the chosen domain; implement both.

15. Applied Data Analytics Mini Project

Design and implement a complete end-to-end data analytics solution (choose from: Student Performance Analytics, Retail Sales Analytics, Healthcare Analytics, Weather Analytics, Financial Data Analytics, or Social Media Analytics). The project must include: (a) data collection from at least 2 sources (CSV + API or two CSV files); (b) data cleaning report (missing values, duplicates, outliers — before and after statistics); (c) exploratory analysis with minimum 8 visualisations across at least 4 chart types; (d) a Plotly interactive dashboard with dropdown filter; (e) a time-series or correlation analysis; (f) a written insights report (minimum 5 data-driven insights with supporting charts); (g) an AI usage log documenting every AI tool used, the prompts given, and whether the output was accepted, modified, or rejected. Present in a 10-minute demo with Q&A.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments
L1–L2	Remember / Understand	**CO1**	Define DataFrame, Series, dtype; describe Pandas read_csv parameters and NumPy operations.
L3	Apply	**CO1, CO2**	Load data from APIs; apply groupby-transform; compute rolling statistics on time-series.
L4	Analyse	**CO1, CO3**	Analyse distribution shape; compute correlation coefficients; compare chart types.
L5	Evaluate	**CO2, CO3**	Evaluate cleaning strategy per column; assess visualisation appropriateness.
L6	Create	**CO4, CO5**	Design and present a complete analytics solution with dashboard, insights report, and AI usage log.

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	2	2	3	–	–	–	1	2	–
CO2	L5	3	3	2	2	3	–	–	–	1	2	–
CO3	L5	3	3	2	3	3	1	–	–	1	2	–
CO4	L6	3	3	3	3	3	2	–	–	2	3	1
CO5	L6	3	3	3	3	3	2	1	–	2	3	1

CO-PO Correlation Highlights	Strong PO5 (Modern Tools) through Pandas, NumPy, Matplotlib, Seaborn, Plotly, ydata-profiling, and NotebookLM. PO3 (Design/Development) through dashboard design and analytics solution architecture. PO10 (Communication) strongly mapped through analytical report generation, data storytelling, and mini project presentation.

LEARNING RESOURCES

Textbooks

1. TB1: Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and Jupyter, 3rd Edition, O'Reilly Media.
2. TB2: Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media. (Free at jakevdp.github.io/PythonDataScienceHandbook)
3. TB3: Joel Grus, Data Science from Scratch: First Principles with Python, 2nd Edition, O'Reilly Media.

Reference Books

1. Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly Media.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, 3rd Edition, O'Reilly Media. (Selected chapters on data preparation.)
3. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly Media. (Free at greenteapress.com)

Online Resources

- Pandas Documentation – pandas.pydata.org | NumPy Documentation – numpy.org
- Matplotlib Documentation – matplotlib.org | Seaborn Documentation – seaborn.pydata.org
- Plotly Documentation – plotly.com/python |Jupyter Documentation – jupyter.org/documentation
- Kaggle (free datasets and notebooks) – kaggle.com | Google Colab – colab.research.google.com
- Python Official Documentation – docs.python.org/3 |FreeCodeCamp Data Analysis – freecodecamp.org

1. NotebookLM – notebooklm.google.com | NPTEL – Data Science for Engineers – nptel.ac.in

Course Code	ENVIRONMENTAL SCIENCE	L	T	Pr	P	Credits
26AHS00301M	Common to All Branches	2	0	0	0	0

Prerequisites: No specific prerequisites; open to all engineering branches.

Course Objectives

- Create awareness about environmental issues, natural resource management and the need for sustainable development.
- Explain the structure and function of various ecosystems and the significance of biodiversity and its conservation.
- Describe causes, effects and control of various types of pollution and principles of solid waste management.
- Examine social, ethical and legislative issues related to environment and sustainable development in India.
- Understand human population dynamics, environmental health linkages and the role of IT and AI tools in environmental monitoring.

Course Content

UNIT – I	Multidisciplinary Nature of Environmental Studies and Natural Resources	9 Hrs

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	9 Hrs

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	9 Hrs

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	9 Hrs

Core Content	**Social Issues and the Environment:** 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	9 Hrs

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

CO	Course Outcome Statement	Bloom's Taxonomy 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 landslides.	L3–L4 (Apply, Analyze)

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)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessment
L1 – Remember	Recall facts, define terms	**CO1, CO2**	Define ecosystem, biodiversity, pollution; state types of natural resources; name Indian environmental laws
L2 – Understand	Explain, describe, interpret	**CO1–CO5**	Explain ecological succession; describe food chain and food web; describe deforestation effects; explain climate change mechanisms
L3 – Apply	Use knowledge in new contexts	**CO2, CO3, CO5**	Apply SWM hierarchy to a local case; apply conservation strategies to a specific ecosystem; use iNaturalist/QGIS in field work
L4 – Analyze	Examine, compare, differentiate	**CO3, CO4**	Analyze pollution causes and effects for a given scenario; compare in-situ vs. ex-situ conservation; examine enforcement issues in environmental legislation
L5 – Evaluate	Assess, judge, critically evaluate	**CO4**	Evaluate the effectiveness of environmental laws in India; assess climate change mitigation policies; evaluate sustainability of a development project
L6 – Create	Design, propose (extension only)	–	Optional extension: propose an environmental management plan for a local site visited during field work

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 PrjMgmt	PO12 Lifelong Learn.	PSO1 CE Core	PSO2 Profnl Dev.
CO1	L1-L2	2	1	–	–	1	3	3	2	–	2	–	2	2	1
CO	L2-L3	2	2	–	1	2	3	3	2	1	2	–	2	2	2

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 PrjMgmt	PO12 Lifelong Learn.	PSO1 CE Core	PSO2 Profnl Dev.
2															
CO3	L3-L4	2	2	2	2	2	3	3	3	1	2	1	2	2	2
CO4	L4-L5	2	2	2	2	2	3	3	3	1	2	1	3	2	2
CO5	L2-L3	1	1	–	1	3	3	3	2	1	2	–	3	1	2

CO-PO Correlation Highlights	Environmental Science demonstrates its strongest correlation with PO6 (Engineer & Society), PO7 (Environment & Sustainability) and PO8 (Ethics) across all five COs, reflecting the direct societal and environmental focus of the course. PO12 (Lifelong Learning) is well addressed through all COs as the course creates lasting environmental awareness. CO4 and CO5 address PO5 (Modern Tools) through AI/GIS-based environmental monitoring tools. As an audit course, the PSO correlations are secondary, though CO3 and CO4 contribute to PSO1 through pollution engineering and environmental management content relevant to civil infrastructure projects.

Textbooks

1. Erach Bharucha, "Textbook of Environmental Studies for Undergraduate Courses", for University Grants Commission, Universities Press.
2. Palaniswamy, "Environmental Studies", Pearson Education.
3. S. Azeem Unnisa, "Environmental Studies", Academic Publishing Company.
4. K. Raghavan Nambiar, "Text Book of Environmental Studies for Undergraduate Courses as per UGC Model Syllabus", Scitech Publications (India) Pvt. Ltd.

Reference Books

1. Deeksha Dave and E. Sai Baba Reddy, "Textbook of Environmental Science", Cengage Publications.
2. M. Anji Reddy, "Text Book of Environmental Sciences and Technology", BS Publications.
3. J. P. Sharma, "Comprehensive Environmental Studies", Laxmi Publications.
4. J. Glynn Henry and Gary W. Heinke, "Environmental Sciences and Engineering", Prentice Hall of India Private Limited.
5. G. R. Chatwal, "A Text Book of Environmental Studies", Himalaya Publishing House.
6. Gilbert M. Masters and Wendell P. Ela, "Introduction to Environmental Engineering and Science", Prentice Hall of India Private Limited.

Web Resources & Online Platforms

- https://www.coursera.org/learn/python-for-applied-data-science-ai
- https://www.coursera.org/learn/python?specialization=python#syllabus
- NPTEL: Environmental Science – https://nptel.ac.in/courses/124107002
- Google Earth Engine (free cloud geospatial platform) – https://earthengine.google.com
- Global Forest Watch (free deforestation monitoring) – https://www.globalforestwatch.org
- 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
Google Earth Engine (free cloud platform) – real-time satellite data analysis for deforestation, land use and vegetation change detection
QGIS (free, open-source GIS) – environmental mapping, watershed delineation, disaster risk zone mapping, biodiversity hotspot analysis
Python (NumPy, Pandas, Matplotlib, scikit-learn, TensorFlow) on Google Colab (free) – AQI prediction, population growth ML models, climate data analysis
iNaturalist (free, open-source citizen science) – species identification using image recognition during field work; community biodiversity database
eBird (free, Cornell Lab) – bird species recording and mapping during field visits; global bird distribution data
OpenAQ (free, open-source) – global real-time and historical air quality data for ML model training
MaxEnt (free) – species distribution modelling for biodiversity hotspot prediction from occurrence data
Global Forest Watch (free dashboard) – monitoring deforestation and forest cover change using satellite imagery
CMIP6 climate datasets (free, ESGF portal) – global climate model data for Python-based climate change scenario analysis