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.)

INSTITUTION'S INNOVATION COUNCIL (Ministry of Education Initiative)

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

B.Tech.
Electronics and Communication Engineering

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)
Estd. 2001
(Approved by AICTE New Delhi & Affiliated to J N T University, Anantapuramu & Accredited by NAAC)
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)

ELECTRONICS COMMUNICATION AND ENGINEERING

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)

I Year I Semester

S.No.	Course code	Title	L/D	T	Pr	P	Credits
1	26AHS00103T	Physics for Communication Systems	3	0	0	0	3
2	26AHS00102T	Linear Algebra & 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	26AEC0401P	**Department Specific Workshop** Electronics and Electrical Engineering Workshop	0	0	0	2	1
6	26ACS05103T	AI Tools and Applications	1	0	4	0	3
7	26AHS00103P	Physics for Communication Systems Lab	0	0	0	3	1.5
8	26ACS05102P	Computational Thinking and Problem Solving with Python Lab	0	0	0	3	1.5
9	26AHS00101L	NSS/NCC/Scouts and Guides / Community Service	0	0	0	1	0.5
10	26ACS05101M	Cybersecurity Awareness for Engineers	2	0	0	0	0
Total			**13**	**0**	**10**	**9**	**20.5**

I Year II Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AHS00201T	Communicative English	2	0	0	0	2
2	26AHS00204T	Engineering Chemistry for Electronics and Communication Systems	3	0	0	0	3
3	26AHS00203T	Differential Equations & Vector Calculus	3	0	2	0	4
4	26AEE02201T	**Professional core 1** Network Analysis	2	0	2	0	3
5	26AME03201T	Engineering Graphics	1	0	4	0	3
6	26AME03201P	Engineering Workshop	0	0	0	3	1.5
7	26AHS00204P	Engineering Chemistry for Electronics and Communication Systems Lab	0	0	0	3	1.5
8	26AHS00201P	Communicative English Lab	0	0	0	2	1
9	26AHS00201L	Health and wellness, Yoga and Sports	0	0	0	1	0.5
Total			26ACS5201M	**0**	**6**	**9**	**19.5**

II Year I Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AHS00303T	Complex Variables and Probability Theory	3	0	0	0	3
2	26AHS00304T	Biology for Engineers	2	0	0	0	2
3	26AEC04301T	**Engineering Course** • Signals, Systems and Stochastic Processes	2	0	2	0	3
4	26AEC04302T	**Professional core - II** • Electronic Devices and Circuits	2	0	2	0	3
5	26AEC04303T	**Professional core - III** • Digital Circuits Design	2	0	2	0	3
6	26AEC04301P	**Engineering Course Lab** • Signal and Systems Lab	0	0	0	2	1
7	26AEC04302P	**Professional core – II Lab** • Electronic Devices and Circuits Lab	0	0	0	3	1.5
8	26AEC04303P	**Professional core - III Lab** • Digital Circuits Design Lab	0	0	0	3	1.5
9	26AHS00305T	Sustainability and Green Engineering	2	0	0	2	2
10	26ACS05302S	**Skill Enhancement course - I** • Data Structures Using Python	1	0	0	2	2
11	26AHS00302M	Technical Paper Writing and IPR	2	0	0	0	0
		Total	**16**	**0**	**4**	**12**	**22**

B.Tech. II Year II Semester

S.No.	Course code	Title	L	T	Pr	P	Credits
1	26AMB02301T	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	26AEC04401T	**Professional core - IV** • Electronic Circuits Analysis	2	0	2	0	3
4	26AEC04402T	**Professional core - V** • Analog and Digital Communications	2	0	2	0	3
5	26AEE02401T	**Professional core -VI** • Linear Control Systems	2	0	2	0	3
6	26AEC04403T	**Professional core VII** • EM Waves and Transmission Lines	2	0	2	0	3
7	26AEC04401P	**Professional core - IV lab** • Electronic Circuits Analysis Lab	0	0	0	3	1.5
8	26AEC04402P	**Professional core - V Lab** • Analog and Digital Communications Lab	0	0	0	3	1.5
9	26AEC04403S	**Skill Enhancement course-II** • PCB Design and Prototype Development	1	0	0	2	2
10	26AHS00301M	Environmental Science	2	0	0	0	0
		Total	**16**	**0**	**8**	**8**	**22**

	Mandatory Community Service Project Internship of 08 weeks duration during summer vacation

I Year – I Semester

Course Code	PHYSICS FOR COMMUNICATION SYSTEMS	L	T	Pr	P	Credits
26AHS00103T		3	0	0	0	3

Prerequisites: Engineering Mathematics, Basic Electronics, Electromagnetic Theory Fundamentals

Course Objectives

The objective of this course is to:

1. To provide knowledge on wave optics concepts such as interference, diffraction, and polarization with basic AI-assisted optical analysis applications.
2. To introduce lasers, fiber optics, and advanced sensing technologies with simple AI-assisted communication and signal processing techniques.
3. To equip students the fundamentals of quantum mechanics and quantum computing with basic AI-assisted secure communication concepts.
4. To familiarize students with electrical properties of metals and semiconductors using AI-assisted material and conductivity analysis methods.
5. To impart knowledge on electromagnetic field theory and its applications with basic AI-assisted antenna and RFID system concepts.

Course Content

UNIT I: Wave Optics and Quantum Information Processing 9 hrs

Core Content:

Interference: Principle of superposition and coherence, Interference, Types of interference, Interference in thin films (reflection geometry) and Newton's rings. Applications: Determination of wavelength and refractive index.

Diffraction: Introduction to diffraction, Fresnel and Fraunhofer diffraction, Fraunhofer diffraction due to single, double and N-slits (Diffraction Grating), Dispersive power and Resolving power of grating. Applications: CD/DVD data storage, Barcode scanners.

Polarization: Types of polarization, Nicol's prism, Half-wave and Quarter-wave plates.

AI Integration:

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

Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib and scikit-learn).

UNIT II: Quantum Photonics and Advanced Sensing 9 hrs

Core Content:

Lasers: Interaction of radiation with matter, Requisites and conditions for laser action, He-Ne laser, Nd-YAG laser, LiDAR basics. Laser-based data communication, Free-space optical communication, Optical data storage, Photonic computing basics, Applications of lasers.

Fiber Optics: Propagation mechanism, Numerical aperture, Acceptance angle, Types of optical fibres, Classifications and Modes of optical fibres, Attenuation and Losses in optical fibers, V- Number - Distributed Fiber Optic Sensing (DFOS) with AI-Powered Signal Analysis.

AI Integration:

AI-powered signal analysis for distributed fiber optic sensing. Python simulation of laser propagation and fiber optic signal attenuation.

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

UNIT III: Introduction to Quantum Mechanics & Quantum Computing 10 hrs

Core Content:

Introduction to quantum mechanics, de-Broglie concept (matter waves), Heisenberg's Uncertainty Principle, Wave function, Postulates of Quantum Mechanics, Schrödinger's time-independent wave equation, Particle in a one-dimensional potential well, Quantum tunnelling (basic idea).

Introduction to Quantum Computing: Classical bits vs Qubits. Concept of superposition and Entanglement, Basic Quantum Operations: Hadamard gate (idea of superposition), Pauli-X gate. Basic idea of quantum measurement. Applications (qualitative only): AI-Assisted Tunnel Diodes and cold emission of electrons, Basic idea of quantum cryptography and secure communication.

AI Integration: AI-Assisted Analysis of Tunnel Diode Characteristics using Quantum Tunnelling Concepts, AI-Assisted Quantum Key Distribution for Secure Communication

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

UNIT IV: Electrical Properties of Metals and Semiconductors 8 hrs

Core Content:

Classical free electron theory – merits and demerits; Quantum free electron theory – equation for electrical conductivity, Fermi-Dirac statistics, Fermi level and variation with temperature, Classification of solids – metals, insulators and semiconductors.

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

AI Integration: AI-Assisted Prediction of Electrical Conductivity in Metals and Semiconductors, AI-Assisted Carrier Concentration and Conductivity Analysis in Semiconductors.

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

UNIT V: Introduction to EM Theory 9 hrs

Core Content:

Introduction, wave equation in differential form in free space, Vector analysis, Divergence and curl of electric field and magnetic field (static), Gauss' divergence theorem and Stokes' theorem. Derivations of the four Maxwell equations and their significance. Maxwell's equations in vacuum, Energy transport and Poynting Vector. Applications: Radio Frequency Identification (RFIDs) and Antennas.

AI Integration: AI-Assisted RFID-Based Smart Tracking and Identification Systems, AI-Assisted Fault Detection in Antenna Systems,

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

Course Outcomes (COs)

After completion of the course, student will be able to

CO	Course Outcome Statement	Bloom's Level
CO1	Apply the principles of interference, diffraction, polarization, along with basic AI-assisted optical analysis in engineering applications.	L4 (Analyze)
CO2	Analyze the working of lasers, fiber optics, and advanced sensing systems along with basic AI-assisted optical communication and signal processing techniques..	L4 (Analyze) L5 (Evaluate)
CO3	Apply the concepts of quantum mechanics and quantum computing along with basic AI-assisted secure communication applications.	L3 (Apply) L4 (Analyze)
CO4	Analyze the electrical properties of metals and semiconductors, along with basic AI-assisted material characterization and conductivity analysis methods	L4 (Analyze) L5 (Evaluate)
CO5	Apply Maxwell's equations and electromagnetic field concepts along with AI-assisted RFID and antenna-based systems.	L3 (Apply) L4 (Analyze)

CO – PO Articulation Matrix

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

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

Textbooks

1. Avadhanulu, Kshirsagar&Arun Murthy, A Textbook of Engineering Physics, S. Chand, 2019.
2. D. K. Bhattacharya & Poonam Tandon, Engineering Physics, Oxford Press, 2015.

Reference Books

1. B. K. Pandey & S. Chaturvedi, Engineering Physics, Cengage Learning, 2021.
2. Shatendra Sharma &Jyotsna Sharma, Engineering Physics, Pearson, 2018.
3. D. J. Griffiths, Introduction to Quantum Mechanics, Cambridge Press.
4. Saleh & Teich, Fundamentals of Photonics, Wiley.
5. Streetman & Banerjee, Solid State Electronic Devices, Pearson.
6. C. Kittel, Introduction to Solid State Physics, Wiley.

Web Resources & NPTEL Links

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

Recommended Free & Open-Source AI/Computational Tools

- Python (NumPy, SciPy, OpenCV, Matplotlib) – FREE on Google Colab for optics and EM simulations
- IBM Qiskit (free, open-source) – For quantum computing and quantum cryptography simulations
- scikit-learn (free, open-source) – For semiconductor data analysis and signal classification
- OpenEMS (free, open-source) – For electromagnetic field simulation and antenna design
- PhET Interactive Simulations (free) – For wave optics, semiconductors and quantum phenomena

I Year – I Semester

Course Code	LINEAR ALGEBRA AND CALCULUS	L	T	Pr	P	Credits
26AHS00102T	Common to All Branches	3	0	2	0	4

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 Content

UNIT I Matrix Algebra and Linear Systems

Core Content:

Matrices, types of matrices, operations, transpose, inverse, symmetric and skew-symmetric matrices; determinant properties; rank of a matrix; echelon and normal forms; system of linear equations; consistency conditions; Gauss elimination and Gauss-Jordan methods; iterative methods: Jacobi and Gauss-Seidel; Cramer's rule; matrix inversion method; applications to engineering systems.

AI Integration:

Python-based solution of large linear systems using NumPy; visualization of convergence in iterative methods; AI-assisted debugging of matrix algorithms; using ChatGPT/Copilot to explain row operations and rank reduction.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy for symbolic matrix operations; Jupyter Notebook; Scilab.

UNIT II Eigenvalues, Eigenvectors, Diagonalization & Quadratic Forms

Core 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 III Single Variable Calculus and Series Expansions

Core 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 for verification.

UNIT IV Multivariable Differentiation and Optimization

Core 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 V Multiple Integrals and Coordinate Transformations

Core Content:

Double integrals; evaluation over type I and type II regions; change of order of integration; triple integrals; change of variables; polar, cylindrical, and spherical coordinates; area and volume computation; mass, centroid and moment calculations; engineering applications of multiple integration.

AI Integration:

Numerical evaluation of double and triple integrals using SciPy; 2D and 3D region visualization; AI-assisted generation of integration code; interpreting Jacobian as geometric scaling factor in transformed coordinates.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib); Google Colab; SymPy; Jupyter Notebook; 3D plotting tools.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply matrix algebra techniques to solve systems of linear equations and analyse matrix properties relevant to engineering models.	L3 (Apply)
CO2	Analyse 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	Analyse multivariable functions using partial derivatives, Jacobians, gradients, and optimization techniques.	L4 (Analyze)
CO5	Evaluate areas, volumes, and coordinate transformations using multiple integrals in engineering contexts.	L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	2	1	–	–	–	1	2	–	–	2	1
CO2	L4	3	3	3	2	1	–	–	–	1	2	–	–	2	1
CO3	L3	3	2	2	2	1	–	–	–	–	2	–	–	2	1
CO4	L4	3	3	3	3	2	–	–	–	1	2	–	–	2	2
CO5	L5	3	3	2	3	3	–	1	1	2	2	1	–	2	2

Textbooks

1. Grewal B.S., Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2017.
2. Kreyszig E., Advanced Engineering Mathematics, 10th Edition, Wiley, 2018.
3. Thomas G.B., Weir M.D., Hass J., Thomas' Calculus, 14th Edition, Pearson, 2018.

Reference Books

1. Strang G., Introduction to Linear Algebra, 5th Edition, Wellesley-Cambridge Press, 2016.
2. Lay D.C. et al., Linear Algebra and Its Applications, 5th Edition, Pearson, 2015.
3. Zill D.G. and Wright W.S., Advanced Engineering Mathematics, 6th Edition, Jones & Bartlett, 2018.

Web Resources & NPTEL Links

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

I Year – I Semester

Course Code	DESIGN THINKING	L	T	Pr	P	Credits
26ACS05101T	Common to All Branches	1	0	4	0	3

Prerequisites: None

Course Overview

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

Course Objectives

- Introduce students to the philosophy, origin, and relevance of Design Thinking in human-centred problem solving and engineering innovation.
- Develop empathy-driven research skills using interviews, observation, and user personas to define meaningful problem statements.
- Enable ideation using structured creative techniques augmented by AI tools for idea generation and concept selection.
- Equip students with prototyping skills across low-fidelity paper and high-fidelity digital formats using AI-powered design tools.
- Develop user testing, iteration, and pitching competencies — integrating AI throughout the full design cycle.

Course Content

UNIT I — Introduction to Design Thinking

Core 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: IDEO shopping cart, Stanford Mobile Clinic.

AI Integration:

Use AI to explain design thinking in different domains (healthcare, education, agriculture), create a mindmap of the 5-stage process, Watch an IDEO documentary and use AI to summarise key DT principles observed.

Free/Open-Source AI Tools:

ChatGPT / Copilot (free tiers); MindMeister (free) / FreeMind (free open-source); Gemini (free)

UNIT II — Empathise and Define Stages

Core 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:

Create digital empathy maps and POV statements collaboratively, generate diverse user persona profiles for a given problem context, Interview 5 peers about a campus problem; use AI to analyse interview transcripts and extract themes, affinity mapping of qualitative research data

Free/Open-Source AI Tools:

Miro (free plan); Dovetail (free plan); ChatGPT / Gemini (free tiers)

UNIT III	Ideation Stage

Core 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:

Use 'expand my idea' prompts to generate 20 variations of a concept, digital brainstorming and affinity diagramming, generate visual concept sketches from text descriptions, run a 30-minute AI-augmented brainstorming session and compare idea quality with pure human brainstorming

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); IdeaFlip (free) / Miro (free plan); DALL-E 3 via ChatGPT (free credits) for visual concept sketches

UNIT IV	Prototyping

Core 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:

Ceate digital wireframes and interactive mockups, convert hand-drawn sketches to digital wireframes using AI, create simple 3D models for physical prototyping, build a paper prototype for a chosen problem and photograph for digital documentation

Free/Open-Source AI Tools:

Figma (free plan); Uizard (free plan, AI-powered); Tinkercad (free, web-based); ChatGPT / Gemini for idea refinement

UNIT V Testing, Iteration and AI in Design

Core Content:

User Testing Methods: think-aloud protocol, A/B testing, usability testing. Feedback Analysis: affinity mapping of test feedback. 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:

Conduct remote usability tests on Figma prototypes, analyse collected user feedback and generate insights and improvement suggestions, present a final prototype to a panel; use AI to prepare pitch slides in Canva, create a Design Thinking journal documenting each stage with AI tool usage notes

Free/Open-Source AI Tools:

Maze (free plan); Canva (free); ChatGPT / Gemini (free tiers); Figma (free plan)

Course Outcomes (COs)

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

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	1	2	2	1	1	2	1	1	2	2	1	2	1	1
CO2	L3	1	2	3	2	2	3	1	1	3	2	2	2	2	2

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO3	L4	2	3	3	2	3	3	1	1	2	2	2	3	2	2
CO4	L5	2	3	3	3	3	2	1	2	2	2	2	3	2	2
CO5	L6	2	3	3	3	3	2	2	2	3	3	3	3	3	3

Textbooks

1. Brown T., Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation, HarperBusiness, 2019.
2. Lewrick M., Link P. and Leifer L., The Design Thinking Playbook, Wiley, 2018.

Reference Books

1. IDEO.org, The Field Guide to Human-Centered Design, IDEO.org, 2015.
2. Kelley T. and Kelley D., Creative Confidence, Crown Business, 2013.
3. Liedtka J. and Ogilvie T., Designing for Growth, Columbia Business School Press, 2011.

Web Resources

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

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

Prerequisites: None

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 Content

UNIT I Computational Thinking and Programming Basics

Core 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.

AI Integration:

Design and simulate algorithms (e.g., largest of 3 numbers); draw flowcharts; trace algorithm execution; convert pseudocode to Python and verify output.

Free/Open-Source AI Tools:

Flowgorithm (free); Lucidchart (free); Algorithm Visualizer (free online); Python on Google Colab (free); AI tools: ChatGPT / Gemini

UNIT II Decision Making and Looping

Core Content:

Decision control statements: Boolean expressions; if, if-else, if-elif-else, nested if; conditional expressions (ternary operator). Looping statements: while loop, for loop, nested loops; loop control — break, continue, pass; else with loops. Practical problem solving: prime number check, pattern programs using nested loops, menu-driven programs.

AI Integration:

Visualize if-else and loop execution step by step; debug loop-based programs; online coding practice; generate test cases for control flow programs.

Free/Open-Source AI Tools:

Python Tutor (free online); Thonny (free); Replit (free); Google Colab (free); AI tools: ChatGPT / Gemini

UNIT III Strings and Data Structures

Core 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.

AI Integration:

Perform text analysis (sentiment, word frequency) on strings; list and array operations; explain string slicing and list comprehensions with concrete examples.

Free/Open-Source AI Tools:

NLTK (free); TextBlob (free); Pandas (free); NumPy (free); Google Colab (free); AI tools: ChatGPT / Gemini

UNIT IV Functions and Problem Solving

Core 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; applications of functions in problem solving.

AI Integration:

Interactive function experiments; symbolic mathematical expressions; modular program development; trace recursive function calls step by step.

Free/Open-Source AI Tools:

Jupyter Notebook (free); SymPy (free); PyCharm community edition (free); Google Colab (free); AI tools: ChatGPT / Gemini

UNIT V File Handling, Exception Handling, and OOP

Core 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.

AI Integration:

Read and write CSV/Excel files; testing exception handling; explain OOP class design with real-world examples; simple mini web application as a capstone project.

Free/Open-Source AI Tools:

Pandas (free); OpenPyXL (free); Pytest (free); Google Colab (free); AI tools: ChatGPT / Gemini

Course Outcomes (COs)

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

CO – PO Articulation Matrix

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

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

Textbooks

1. Spronck P., Computational Thinking with Python (covers computational thinking, algorithms, and Python programming).
2. Thareja R., Programming in Python, Oxford University Press (primary implementation reference).

Reference Books

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

Web Resources

- Flowgorithm – http://flowgorithm.org
- Python Tutor – https://pythontutor.com
- NPTEL – Problem Solving through Programming in Python – https://nptel.ac.in
- Google Colab – https://colab.research.google.com

ELECTRONICS & ELECTRICAL ENGINEERING WORKSHOP

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC0401P	I B.Tech I Sem	0	0	0	3	1.5	Department Specific Workshop

Prerequisites: Basic Physics and Mathematics (Class XII level); no concurrent prerequisite.

Course Objectives

- Gain hands-on proficiency in workshop tools, measuring instruments, and electronic components through assembly, testing, and theoretical vs. measured value comparison.
- Understand electrical systems and implement verified DC and AC circuits on breadboard and PCB experimentally.
- Measure and interpret electrical parameters — real power, power factor, earth resistance, and energy consumption — using standard instruments such as wattmeter, Megger, and energy meter.
- Apply Python and AI/LLM tools to automate data analysis, simulate circuit behaviour, and validate experimental results through regression and predictive modelling.
- Design a socially relevant mini project using AI/LLM tools and Python/MATLAB for simulation, analysis, and prediction.

List of Activities

Part A – Regular Activities

- Familiarization of commonly used workshop tools like breadboard, soldering iron, relays, switches, connectors, fuses, cutters, pliers, wire stripper, etc.; exercises for proper handling and usage.
- Familiarization with voltmeters, ammeters, multimeter, LCR meter, power supplies, CRO/DSO, function generator, and frequency counter; perform basic measurement experiments.
- Identification of components (R, L, C, diodes, transistors, ICs) — type, symbol, colour coding, and package; testing and comparison of theoretical vs. measured values.
- Study of fundamental electrical concepts — voltage, current, power, AC/DC, single-phase, three-phase; understanding the working principles of transformers, generators, and motors.
- Calculate energy consumption of domestic appliances and/or a premises using standard formulas; record and tabulate energy usage data for a given set of loads.
- Measure real power and power factor of a single-phase AC circuit using a wattmeter; tabulate readings for different load conditions.
- Measure earth resistance of the laboratory grounding electrode using a Megger instrument; tabulate readings and compare with permissible safety limits.

- Assemble any two circuits — one with a DC source and one with an AC source — on a breadboard and test their functionality; observe and record output waveforms or measured parameters.
- Assemble any two circuits — one with a DC source and one with an AC source — on a PCB and test their performance; compare measured results with expected values.
- Design and implement an electronic or hobby circuit with a societal application (e.g., energy monitoring system, sensor-based automation, fault detection system); demonstrate working functionality.

Part B – AI Integration Lab Exercises

- Any **five** activities must be implemented using appropriate AI/LLM tools and/or Python/MATLAB-based simulations.

Activity 1

a) Use AI/LLM tools to query tool datasheets and manufacturer specifications; generate an interactive Q&A to identify tool functions, correct handling postures, and safety hazards for each workshop tool.

b) Use LLM-generated safety checklists to create a Python-based scoring tool that evaluates a student's handling procedure against a standard rubric.

Activity 2

a) Use AI/LLM tools to explain the operating procedures of each instrument (CRO/DSO, LCR meter, function generator) and to interpret measurement results — including scale reading, range selection, and source of error.

b) Use Python (NumPy/Matplotlib) to plot measured voltage and current data; apply linear regression to verify Ohm's Law and estimate percentage error from ideal values.

Activity 3

a) Use AI/LLM tools for AI-assisted component identification — describe a component by its markings or package and receive datasheet summary, pin-out, and typical application; understand colour-code decoding for resistors and capacitors.

b) Use Python (Scikit-learn) regression to estimate device parameters (static resistance, capacitance, inductance) from LCR meter readings and compare with datasheet nominal values.

Activity 4

a) Use AI/LLM tools to relate fundamental electrical concepts (voltage, current, power factor, single/three-phase systems) to real-world applications — household wiring, industrial motors, and transformer ratings; generate concept maps.

b) Use Python to simulate single-phase and three-phase phasor diagrams (Matplotlib); model transformer turns ratio and calculate secondary voltage for given primary inputs.

Activity 5

a) Use AI/LLM tools to explain energy billing calculations, units (kWh), tariff slabs, and the impact of power factor on energy cost; generate a structured energy audit report for a given premises.

b) Use Python (Google Colab/Scikit-learn) to model and predict monthly energy consumption using linear regression; input appliance wattage and daily usage hours as features.

Activity 6

a) Use AI/LLM tools to explain power factor, reactive power compensation, and the effect of capacitive/inductive loads; auto-generate a power quality summary from wattmeter readings.

b) Use Python/MATLAB to plot real power, reactive power, and power factor against load variation; apply regression to model the relationship between load impedance and power factor.

Activity 7

a) Use AI/LLM tools to explain earth resistance standards (IS 3043/IEEE 80), the fall-of-potential method used by Megger, and the significance of low earth resistance for human safety; auto-generate a safety compliance report.

b) Use Python to analyse tabulated Megger readings; flag values exceeding safety thresholds using a rule-based classifier and plot resistance vs. electrode depth.

Activity 8

a) Use AI/LLM tools to interpret the circuit schematic, suggest component values, and debug functional faults based on described symptoms (e.g., no output, distorted waveform, wrong DC level).

b) Use Python/MATLAB to simulate both the DC and AC circuits before hardware implementation; compare simulated waveforms or voltage levels with measured breadboard results.

Activity 9

a) Use AI/LLM tools for PCB layout guidance — component placement rules, trace width calculation for current capacity, and design-rule-check (DRC) criteria; generate a bill of materials (BOM) from the schematic.

b) Validate the PCB design using MATLAB/Python simulation prior to fabrication; compare simulated and measured performance and quantify deviation using percentage error analysis.

Activity 10

a) Use AI/LLM tools throughout the project lifecycle — problem statement framing, circuit design justification, component selection rationale, testing strategy, and final report generation including abstract, block diagram description, and conclusion.

b) Integrate Python/MATLAB for data acquisition, analysis, or prediction within the project (e.g., regression-based energy forecasting, threshold-based fault classifier, sensor signal processing); include model training, validation, and result visualisation.

AI Integration	AI integration covers five selected activities: (i) Python-based tool safety scoring; (ii) Ohm's Law verification via NumPy/Matplotlib linear regression; (iii) LCR meter data regression for component parameters; (iv) Python phasor diagrams and transformer modelling; (v) linear regression-based energy prediction — each reinforcing the link between physical experimentation and computational analysis.

Free / Open-Source AI Tools: Python (NumPy, Matplotlib, Scikit-learn) – Google Colab (free); LTspice / Falstad Circuit Simulator (free) – circuit simulation; Virtual Labs – IIT (free) – online electrical experiments.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Identify, handle, and test workshop tools, instruments, and electronic components; assemble and verify DC and AC circuits on breadboard and PCB using appropriate measuring instruments.	L3 (Apply)
CO2	Calculate electrical energy consumption and power factor; implement and validate circuits through simulation and hardware, comparing experimental results with theoretical values.	L3–L4 (Apply, Analyse)
CO3	Apply AI/LLM and/or Python/MATLAB tools for data analysis, simulation, and report generation; design and demonstrate a mini project with societal application integrating AI and hardware.	L5–L6 (Evaluate, Create)
CO4	Measure earth resistance, real power, and power factor using standard instruments (Megger, wattmeter); compare readings against IS/IEEE permissible limits and interpret results.	L4 (Analyse)
CO5	Automate measurement data processing, energy prediction, and fault detection using Python (NumPy/Matplotlib/Scikit-learn); evaluate model accuracy using appropriate metrics.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	State Ohm's law; define power factor; recall colour codes for resistors; state IS 3043 earth resistance limit
L2 – Understand	Explain procedures, describe instruments	CO1–CO5	Explain difference between CRO and DSO; describe AC vs DC circuits; explain breadboard vs PCB assembly
L3 – Apply	Conduct activities, record, compute	CO1, CO2	Perform energy audit calculation; assemble DC circuit on breadboard; measure power factor with wattmeter
L4 – Analyse	Compare, verify, interpret results	CO2, CO4	Compare measured vs theoretical resistance; analyse earth resistance against safety limits; verify Ohm's Law by regression
L5 – Evaluate	Assess accuracy, evaluate models	CO4, CO5	Evaluate % error in component values; assess Scikit-learn model R^2 for energy prediction; evaluate PCB vs breadboard performance
L6 – Create	Automate, design, demonstrate	CO3, CO5	Code Python energy predictor; design a fault detection circuit; develop AI-assisted BOM generator for PCB project

CO – PO / PSO Articulation Matrix (NBA)

Course Code	AI TOOLS AND APPLICATIONS	L	T	Pr	P	Credits

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

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Prob. Analysis	PO3 Design/ Dev.	PO4 Invest.	PO5 Modern Tools	PO6 Eng. & Society	PO7 Env. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgmt	PSO1 EEE Core	PSO2 Profnl Dev.
CO1	L3	3	2	2	3	2	2	1	2	2	2	–	3	2
CO2	L3–L4	3	3	3	3	2	1	–	1	2	2	–	3	2
CO3	L4	3	3	3	3	3	2	1	1	3	3	1	3	3
CO4	L4	2	3	2	3	3	2	1	2	2	2	–	2	2
CO5	L5–L6	2	3	3	3	3	1	1	1	2	3	1	2	3

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, Matplotlib, Scikit-learn) on Google Colab (free) – energy prediction, Ohm's Law regression, component parameter estimation
Scikit-learn (free, open-source) – linear regression and rule-based classifiers for energy forecasting and fault detection
LTspice / Falstad Circuit Simulator (free) – pre-hardware simulation of breadboard and PCB circuits
Virtual Labs – IIT (free): https://vlab.co.in – online simulation of electrical measurement experiments
LibreOffice Calc (free, open-source) – standardised data recording templates for all activities

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO4
AI Lab Exercises (Python / LLM)	2 exercises	10	CO3, CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every activity at the next session. AI exercises to be submitted as Google Colab .ipynb notebooks.

I Year – I Semester

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–**	Use AI tools to create presentations, research summaries,

		CO3	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 Semester

PHYSICS FOR COMMUNICATION SYSTEMS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AHS00103P	I B.Tech I Sem	0	0	0	3	1.5	

Prerequisites: Basic Physics and Mathematics (Class XII level); no concurrent prerequisite.

Course Objectives

- Determine optical properties of materials — dispersive power, radius of curvature, and wavelength of light — using prism, Newton's rings, and diffraction grating experiments.
- Investigate quantum and laser phenomena by estimating Planck's constant through the photoelectric effect and measuring laser wavelength using diffraction.
- Study the electrical, magnetic, and semiconductor properties of materials — including B–H curves, Hall effect, energy gap, resistivity, and ferroelectric behaviour.
- Measure electromagnetic quantities such as self-inductance, magnetic field along a coil axis, and Hall voltage using standard instruments and experimental setups.
- Apply Python and AI/LLM tools to automate data analysis, simulate physical phenomena, and validate experimental results through computational methods.

List of Experiments

Part A – Regular Experiments

- Determination of the dispersive power of the prism
- Determination of the thickness of a thin object by the wedge method
- Determination of the radius of curvature of the lens by Newton's rings
- Determination of wavelengths of various colours of mercury spectrum using diffraction grating in the normal incidence method
- Determination of wavelength of a LASER using diffraction grating
- Estimation of Planck's constant using the photoelectric effect
- Determination of e/m of an electron by J.J. Thomson method
- Determination of the Curie temperature and dielectric constant of a ferroelectric material
- Study the variation of B versus H by magnetizing the magnetic material (B–H curve)
- Determination of the resistivity of a semiconductor by the four-probe method
- Determination of the energy gap of a semiconductor using p–n junction diode
- Determination of Hall voltage and Hall coefficient of a given semiconductor using Hall effect
- Verification of the inverse square law for a magnetic field using a Hall probe
- Measurement of self-inductance of a coil using LCR meter and the resonance method

- Determination of magnetic field along the axis of a current-carrying circular coil by Stewart & Gee's Method

Part B – AI Integration Lab Exercises

- Any **three** of the following AI exercises must be completed using Python / Google Colab or appropriate AI/LLM tools.
- Optics Data Automation (Python / Matplotlib): input raw angle-of-deviation data from the prism experiment; fit a polynomial to the deviation vs. angle curve; extract the angle of minimum deviation and compute dispersive power; overlay the fitted curve on experimental data points.
- Fringe Analysis using OpenCV (Python / Google Colab): import Newton's ring or wedge-fringe images; apply Gaussian blur and Canny edge detection to locate fringe centres; compute fringe diameters automatically; calculate radius of curvature or thickness and compare with manual measurements.
- Planck's Constant Estimation (Python / NumPy): input stopping potential vs. frequency data from the photoelectric experiment; apply linear regression (V_s vs. ν); extract slope to compute h and y-intercept to find work function; plot the regression line and quantify percentage error against the accepted value 6.626×10^{-34} J·s.

AI Integration	AI integration adds three exercises after the regular experiments: (i) Python automation of prism deviation curve fitting and dispersive power extraction using Matplotlib/NumPy; (ii) OpenCV-based fringe image analysis for Newton's rings and wedge method — automating diameter measurement and radius of curvature computation; (iii) linear regression on photoelectric effect data to extract Planck's constant and work function — reinforcing the connection between the physical experiment and quantum theory.

Free / Open-Source AI Tools: Python (NumPy, SciPy, OpenCV, Matplotlib) – Google Colab (free); scikit-learn (free) – semiconductor data analysis; PhET Interactive Simulations (free) – wave optics and quantum phenomena; VLabs IIT (free) – online physics experiments.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Determine dispersive power of a prism, thickness by wedge method, radius of curvature by Newton's rings, and wavelengths of mercury spectrum and LASER using diffraction grating; record, tabulate, and interpret optical measurement data.	L3 (Apply)
CO2	Estimate Planck's constant from photoelectric effect data; determine e/m by J.J. Thomson method; measure Curie temperature and dielectric constant of a ferroelectric material; relate results to quantum and solid-state physics concepts.	L3–L4 (Apply, Analyse)
CO3	Trace B–H curves of magnetic materials; measure resistivity by four-probe method; determine energy gap of a semiconductor using	L4 (Analyse)

	p–n junction diode; analyse and compare experimental values with theoretical/standard values.	
CO4	Determine Hall voltage and Hall coefficient; verify the inverse square law for magnetic fields; measure self-inductance using LCR meter and resonance method; determine magnetic field along a coil axis by Stewart & Gee's method; evaluate results against theoretical predictions.	L4–L5 (Analyse, Evaluate)
CO5	Automate optical fringe analysis using OpenCV; extract Planck's constant by regression on photoelectric data; perform dispersive power curve fitting with Matplotlib/NumPy; evaluate computational model accuracy using percentage error and R^2.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	State Snell's law; define dispersive power; recall formula for energy gap; state Hall effect; define Curie temperature
L2 – Understand	Explain experiments, describe setups	CO1–CO5	Explain formation of Newton's rings; describe four-probe method; explain principle of diffraction grating; describe J.J. Thomson experiment
L3 – Apply	Conduct experiments, record, compute	CO1, CO2	Perform prism experiment and compute dispersive power; record fringe diameters and calculate radius of curvature; measure stopping potential vs. frequency
L4 – Analyse	Compare, verify, interpret results	CO2–CO4	Compare experimental e/m with standard value; analyse B–H curve hysteresis; compare semiconductor energy gap with standard values; interpret Hall coefficient sign
L5 – Evaluate	Assess accuracy, evaluate models	CO4, CO5	Evaluate % error in Planck's constant; assess regression R^2 for photoelectric data; evaluate OpenCV fringe diameter accuracy vs. manual measurement
L6 – Create	Automate, develop, propose	CO5	Code Python prism curve fitting; develop OpenCV fringe analysis script; write regression-based Planck's constant estimator; propose improved experimental setup

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Prob. Analysis	PO3 Design/ Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Eng. & Society	PO7 Env. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgmt	PSO1 ECE Core	PSO2 Profnl Dev.
CO1	**L3**	3	2	2	3	2	1	1	1	2	2	–	3	2
CO2	**L3–L4**	3	3	3	3	3	1	–	1	2	2	–	3	2
CO3	**L4**	3	3	2	3	2	1	1	1	2	2	–	3	2
CO4	**L4–L5**	3	3	3	3	3	1	1	1	2	2	–	2	3
CO5	**L5–L6**	2	3	3	3	3	1	1	1	2	3	1	2	3

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, SciPy, OpenCV, Matplotlib) on Google Colab (free) – optics automation, fringe analysis, photoelectric regression, EM simulations
scikit-learn (free, open-source) – semiconductor data analysis and signal classification
OpenEMS (free, open-source) – electromagnetic field simulation and antenna design
PhET Interactive Simulations (free) – wave optics, semiconductors, and quantum phenomena: phet.colorado.edu
Virtual Labs – IIT (free): https://iitb.vlabs.co.in/discipline.html?discipline=Physical_Sciences – online simulation of optics, semiconductor and EM experiments
HyperPhysics (free) – concept reference for optics, quantum and EM theory: hyperphysics.phy-astr.gsu.edu
LibreOffice Calc (free, open-source) – standardised data recording templates for all experiments

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI Lab Exercises	2 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI exercises to be submitted as Google Colab .ipynb notebooks.

Reference Books

- M. N. Avadhanulu& P. G. Kshirsagar, *A Textbook of Engineering Physics,* S. Chand & Company, 2019.
- H. K. Malik & A. K. Singh, *Engineering Physics,* Tata McGraw-Hill, 2010.
- R. K. Gaur & S. L. Gupta, *Engineering Physics,*Dhanpat Rai Publications, 2012, 8th Edition.
- B. K. Pandey & S. Chaturvedi, *Engineering Physics,* Cengage Learning, 2012.

I Year I Sem

COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26ACS05102P	I Year I Sem	0	0	0	3	1.5	Common to All Branches

Lab Course Outcomes

- Design algorithms and flowcharts for simple problems; write and execute basic Python programs using variables, data types, and operators.
- Implement programs using if-else, nested if, switch (match), while, and for loops to solve decision-making and iterative problems.
- Apply string operations and Python built-in data structures — lists, tuples, and sets — to solve data manipulation problems.
- Write modular programs using user-defined functions, recursion, lambda expressions, and dictionaries.
- Implement programs using file handling, exception handling, and basic object-oriented programming concepts in Python.

List of Experiments

Unit I – Computational Thinking and Programming Basics

1. Algorithm Design, Flowcharts, and Python Basics

Draw flowcharts using Flowgorithm for three problems: (a) find the largest of two numbers; (b) check whether a number is even or odd; (c) compute the factorial of a given number. For each flowchart, trace through the logic manually with 3 sample inputs before coding. Write equivalent Python programs for all three; run them in Thonny or Google Colab and verify that the output matches the manual trace. Use Python Tutor to visualize the variable state at each step for the factorial program.

2. Data Types, Operators, and Type Conversion

Write Python programs to: (a) declare variables of all primitive data types (int, float, complex, bool, str) and print their type using type(); (b) demonstrate all operator categories — arithmetic, relational, logical, bitwise, and assignment operators — with one example each; (c) demonstrate implicit and explicit type conversion — int to float, float to int, int to str, str to int using int(), float(), str(); (d) print a formatted bill: accept item name, quantity, and price as input; compute and display total using f-string formatting with two decimal places.

3. Input, Output, and Expression Evaluation

Write Python programs to: (a) accept the radius of a circle; compute and display area (πr^2) and circumference ($2\pi r$) using the math module; (b) accept three subject marks; compute total, percentage, and grade (O $\geq$ 90, A+ $\geq$ 80, A $\geq$ 70, B+ $\geq$ 60, B $\geq$ 50, C $\geq$ 40, F $<$ 40); print a formatted result card; (c) accept two numbers and an operator (+, −, ×, ÷) as input; evaluate the expression using eval(); handle division by zero with a simple if condition; (d) demonstrate operator precedence: compute 5 mixed expressions — predict the result manually, then verify by running the program.

Unit II – Decision Making and Looping

4. Decision Control — if, elif, else, and match

Write Python programs to: (a) accept a year and check whether it is a leap year using nested if — a year is a leap year if divisible by 400, or divisible by 4 but not by 100; test with 2000, 1900, 2024, 2023; (b) accept a character and classify it as uppercase letter, lowercase letter, digit, or special character using if-elif-else; (c) accept a day number (1–7) and print the day name using match-case (Python 3.10+); add a default case for invalid input; (d) implement a simple ATM menu using nested if — display options (Check Balance, Deposit, Withdraw, Exit); perform the selected operation; display appropriate error messages for invalid PIN or insufficient balance.

5. Loops — while, for, and Nested Loops

Write Python programs to: (a) print multiplication table of a number entered by the user using a for loop (1 to 10); (b) print all prime numbers between 1 and 100 using a while loop with a flag variable — also count and display the total number of primes found; (c) print five patterns using nested for loops: right-angled triangle (stars), inverted triangle, number pyramid, Floyd's triangle, and diamond pattern — accept the number of rows as input; (d) implement a number guessing game — generate a random number between 1 and 50 using random.randint(); allow a maximum of 7 attempts; print 'Too High' or 'Too Low' as hints; use break when the correct number is guessed.

6. Loop Control and Application Programs

Write Python programs to: (a) compute the sum of digits, reverse, and check if palindrome for a number entered by the user — use a while loop extracting digits using % and //; test with 121, 12321, 1234; (b) compute the sum of the Fibonacci series up to N terms using a for loop; also print the series itself; (c) accept a positive integer N and compute: sum of natural numbers, sum of squares, and sum of cubes — verify sum of cubes formula: $[N(N+1)/2]^2$; (d) implement a simple calculator using a while loop with a menu — loop continues until the user selects Exit; use continue to handle invalid menu choices without breaking the loop.

Unit III – Strings and Data Structures

7. String Operations and Methods

Write Python programs to: (a) accept a string and perform: count vowels and consonants, reverse the string, check palindrome, count words, convert to title case — display all results; (b) demonstrate 10 built-in string methods: upper(), lower(), strip(), replace(), split(), join(), find(), count(), startswith(), endswith() — with one practical example each; (c) accept a sentence and: find the longest word, count frequency of each word (store in a dictionary), print words in alphabetical order; (d) implement a simple Caesar cipher — accept a message and a shift value (1–25); encrypt each letter by shifting it; display encrypted message; write a decrypt function to recover the original message.

8. Lists and Tuples

Write Python programs to: (a) create a list of 10 integers; perform: insert at a given position, delete by value, find maximum and minimum without using built-in max/min, sort using

Bubble Sort, search using linear search — display the list after each operation; (b) demonstrate list slicing: extract first half, second half, every alternate element, and reverse using slice notation; compare with string slicing; (c) store student records as a list of tuples — each tuple: (roll_no, name, marks); sort by marks (descending) using sorted() with a lambda key; print rank list; find the student with highest marks using max() with a key; (d) demonstrate list comprehension: generate squares of even numbers from 1 to 20; filter words longer than 4 characters from a word list; create a multiplication table as a 2D list using nested comprehension.

9. Sets and Applications

Write Python programs to: (a) create two sets A and B from user input; compute and display: union, intersection, difference, and symmetric difference; verify De Morgan's laws for sample values; (b) use a set to remove duplicates from a list of 20 student roll numbers (with deliberate duplicates); compare list size before and after; print unique roll numbers in sorted order using sorted(set()); (c) demonstrate frozen sets: create a frozenset of prime numbers up to 20; attempt to add an element (show TypeError); use it as a dictionary key (demonstrate immutability advantage); (d) implement a program to find: students who passed all three subjects, students who failed at least one, students who passed exactly two — store each class in a set and use set operations.

Unit IV – Functions and Problem Solving

10. Dictionaries and Applications

Write Python programs to: (a) create a student marks dictionary {name: [marks_list]}; compute and display average marks, highest scorer, and students who scored below 50; add a new student, update an existing student's marks, delete a student — show the dictionary after each operation; (b) implement a word frequency counter — accept a paragraph as input; split into words; build a frequency dictionary using a for loop; print the top 5 most frequent words sorted by count descending; (c) demonstrate all dictionary methods: keys(), values(), items(), get(), update(), pop(), setdefault(), copy() — with one example each; (d) build a simple phone book — store contacts as {name: phone}; support: add contact, search by name, delete contact, display all contacts sorted alphabetically — use a while loop menu.

11. User-Defined Functions and Scope

Write Python programs to: (a) write 5 utility functions: is_prime(n), is_palindrome(s), factorial(n), gcd(a,b), power(base, exp) — call each with 3 test inputs and verify results; (b) demonstrate all four argument types: positional, default, keyword, and variable-length (*args for sum of any count, **kwargs for student info display) — write one function for each and test; (c) demonstrate local vs. global scope: write a program with a global counter variable modified inside a function using the global keyword; contrast with a local variable of the same name; (d) write a function find_stats(numbers) that accepts a list and returns mean, median, mode, and standard deviation without using the statistics module — test on [4, 7, 13, 2, 7, 4, 7, 1].

12. Recursion and Lambda Functions

Write Python programs to: (a) implement recursive functions for: factorial(n), fibonacci(n), sum_of_digits(n), binary_search(arr, target, low, high), and power(base, exp) — for each, use Python Tutor to trace the call stack for a small input and count the total recursive calls; (b) compare iterative vs. recursive factorial for n = 5, 10, 15, 20 on execution time using time.time(); document when recursion becomes slower; (c) implement lambda functions for: square, cube, even/odd check, Celsius-to-Fahrenheit, and concatenate two strings — use each with map() and filter() on a list of 10 elements; (d) write a higher-order function apply_twice(f, x) that applies function f to x twice; test with square, double, and increment functions; demonstrate that lambda functions work as arguments.

Unit V – File Handling, Exception Handling, and OOP

13. File Handling

Write Python programs to: (a) create a text file and write 10 lines of student records (roll number, name, marks); read and display all lines; append 5 more records; count total lines; display only records where marks > 60; (b) copy the contents of one text file to another; count total characters, words, and lines in the source file; find and replace a specific word (e.g., 'Python' → 'programming') and write the result to a new file; (c) read a CSV file (student data) using the csv module; compute average marks per subject; write a summary report CSV with student name and grade; (d) create a student marks file; sort all records by marks in descending order (read all, sort in memory, rewrite); demonstrate the difference between read modes: 'r', 'w', 'a', 'r+' with examples.

14. Exception Handling

Write Python programs to: (a) write a program that accepts two numbers and performs division; handle ZeroDivisionError, ValueError (non-numeric input), and TypeError in separate except blocks; use a finally block to print 'Operation complete' regardless of outcome; (b) implement a robust file reader that handles FileNotFoundError and PermissionError; use try with multiple except clauses; after handling each exception, print a meaningful user-friendly error message rather than a traceback; (c) define a custom exception class InsufficientFundsError(Exception) with a message attribute; implement a BankAccount class with deposit() and withdraw() methods — withdraw() raises InsufficientFundsError if balance is insufficient; (d) demonstrate exception chaining (raise X from Y) — catch a ValueError and re-raise as a custom ApplicationError with the original exception attached; use raise without arguments in a re-raise scenario.

15. Object-Oriented Programming — Class Design Capstone

Design and implement a Student Management System using OOP: (a) create a Student class with attributes: roll_no, name, marks (list of 5 subjects) and methods: compute_percentage(), get_grade() (O/A+/A/B+/B/C/F), display() — create 5 Student objects with different data; (b) add a class variable total_students that increments with each object creation; add a class method get_total() and a static method validate_marks(marks) that returns True if all marks are between 0 and 100; (c) write a find_topper(students) function that accepts a list of Student objects and returns the one with the highest percentage; sort the list by percentage using sorted() with a lambda key; (d) write all student records to a file using the write() method; read them back and reconstruct Student objects; print the final class report in tabular format using f-strings with column alignment.

AI Tools Integration (Lab)	Use Thonny IDE (beginner-friendly, built-in debugger) or Google Colab as the primary lab environment. Use Python Tutor (pythontutor.com) to visualize variable states, recursive call stacks, and list/dictionary operations

	— mandatory for Experiments 3, 12, and 15. Use Flowgorithm to design flowcharts before writing code in Experiments 1 and 4. Use ChatGPT or Claude to explain error messages when a program fails — read the explanation, fix the error yourself, and document both the error and the fix. Do NOT copy AI-generated code directly; use AI to understand concepts and debug, then write the solution independently.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Design algorithms and flowcharts for simple problems; write and execute basic Python programs using variables, data types, and operators; evaluate expressions and trace program execution step by step.	L3 (Apply)
CO2	Implement programs using decision-making constructs (if-else, nested if, match-case) and looping structures (while, for, nested loops) to solve decision-making and iterative problems.	L3 (Apply)
CO3	Apply string operations and Python built-in data structures — lists, tuples, and sets — to solve data manipulation, searching, and sorting problems.	L3–L4 (Apply, Analyze)
CO4	Write modular programs using user-defined functions, recursion, lambda expressions, higher-order functions, and dictionaries to decompose and solve complex problems.	L4 (Analyze)
CO5	Implement programs using file handling, exception handling, and basic object-oriented programming concepts; design and test a multi-class Python application.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	Define algorithm, flowchart, data type; state Python operator precedence rules; recall recursion definition; state exception hierarchy
L2 – Understand	Explain programs,	CO1–CO5	Explain difference between while and for loops; describe list vs. tuple vs. set; explain local vs. global scope;

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
	describe logic		describe OOP concepts
L3 – Apply	Write programs, execute, compute	CO1, CO2	Write a factorial program; implement Bubble Sort; create a Caesar cipher; implement ATM menu with nested if
L4 – Analyze	Compare, verify, trace	CO2–CO4	Compare iterative vs. recursive factorial on execution time; trace call stack in Python Tutor; analyze set operations for student pass/fail classification
L5 – Evaluate	Assess programs, validate results	CO4, CO5	Evaluate find_stats() accuracy without statistics module; assess exception handling robustness; evaluate OOP design choices for Student Management System
L6 – Create	Design and build programs	CO5	Design and implement Student Management System with file I/O; write a multi-class OOP application; build a full exception-handling bank account system

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investigation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 PrjMgmt	PO12 Lifelong Learn.	PSO1 CS Core	PSO2 Profnl Dev.
CO1	**L3**	3	2	1	2	3	1	–	1	1	2	–	2	3	2
CO2	**L3**	3	2	1	2	3	1	–	1	2	2	–	2	3	2
CO3	**L3–L4**	3	3	2	2	3	1	–	1	2	2	–	2	3	2
CO4	**L4**	3	3	3	3	3	1	–	1	2	2	1	3	3	3
CO5	**L5–L6**	2	3	3	3	3	1	1	1	2	3	1	3	2	3

Recommended Free & Open-Source AI / Computational Tools

- Thonny IDE (free) – thonny.org: beginner-friendly Python IDE with a built-in step debugger; primary lab environment for all experiments.
- Google Colab (free) – colab.research.google.com: cloud Jupyter notebook; enables GPU-accelerated runs with no local installation.
- Python Tutor (free) – pythontutor.com: visualize variable states, recursive call stacks, list and dictionary operations — mandatory for Experiments 3, 12, and 15.
- Flowgorithm (free) – flowgorithm.org: drag-and-drop flowchart tool with direct Python code generation; mandatory for Experiments 1 and 4.
- ChatGPT / Claude (free tier) – for explaining error messages and debugging hints; students must read, understand, and independently fix errors — do NOT copy AI-generated code.
- Replit (free) – replit.com: browser-based Python IDE; useful for collaborative pair programming sessions.
- Python Official Docs – docs.python.org/3: authoritative language reference; use for verifying built-in function behaviour.

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI Lab Exercises (Python Tutor trace + flowchart)	2 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. Python Tutor traces (screenshots / exported links) to be pasted into the lab record for mandatory experiments. AI exercises (Experiments 3, 12, 15) to be submitted as Google Colab .ipynb notebooks.

Learning Resources

Textbooks

- TB1: Pieter Spronck, Computational Thinking with Python — algorithms and Python foundations.
- TB2: Reema Thareja, Programming in Python, Oxford University Press — primary implementation reference for all five units.

Reference Books

- Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly. (Free at greenteapress.com)
- Eric Matthes, Python Crash Course, 3rd Edition, No Starch Press.

Online Tools and Resources

- Python Tutor – pythontutor.com | Flowgorithm – flowgorithm.org
- Thonny IDE – thonny.org | Google Colab – colab.research.google.com
- Python Official Documentation – docs.python.org/3
- NPTEL – Problem Solving through Programming in Python – nptel.ac.in
- W3Schools Python Tutorial – w3schools.com/python | Replit – replit.com

I Year – I Semester

NSS / NCC / SCOUTS & GUIDES / COMMUNITY SERVICE

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AHS00101L	I Year I Sem	0	0	0	1	**0.5**	Common to All Branches

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	Course Outcome Statement	Bloom's Level
CO1	Understand the philosophy, history, and objectives of NSS/NCC/Scouts & Guides and community service programmes.	L1–L2 (Remember / Understand)
CO2	Demonstrate leadership, teamwork, and organisational skills through camp/parade/drill activities.	L3 (Apply)
CO3	Plan and execute community service projects addressing social issues like health, hygiene, literacy, and environment.	L3–L4 (Apply / Analyse)
CO4	Apply first aid, disaster-preparedness, and civic responsibilities in real-world scenarios.	L3 (Apply)
CO5	Develop civic sense, national integration, and sustainable community-development values.	L5 (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

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 2 – Leadership, Skill Development, Camping and Disaster Management

Topics

- Leadership development – theories and models (situational, transformational)
- Team building, communication & conflict resolution
- Parade, drill & physical training basics
- Map reading, navigation and field craft (NCC)
- Camp organisation – planning, logistics, safety protocols
- Adventure activities: trekking, rock-climbing, swimming (awareness)
- Disaster management – concepts, types, roles of NSS/NCC volunteers
- First Aid – CPR, wound management, fractures, snake bite, burn

UNIT 3 – Community Projects, Social Awareness and Service Documentation

Topics

- Community needs assessment – participatory methodology
- Project planning: Swachh Bharat, Pulse Polio, Blood Donation camps
- Literacy campaigns and awareness drives (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

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	–	–	–	–	–

Text Books & References

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

I Year I Sem

Course Code	CYBERSECURITY AWARENESS FOR ENGINEERS	L	T	Pr	P	Credits
26ACS05101M	Common to All Branches	2	0	0	0	0

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.
- Familiarise students with emerging cybersecurity concerns in cloud, digital platforms, and AI-enabled environments.

Course Content

UNIT I Introduction to Cyberspace and Cybersecurity

Core Content:

Introduction to cyberspace and cybersecurity — evolution of information systems and the 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 behaviour. Cyber hygiene practices.

AI Integration:

Use AI/LLM tools to explain the CIA Triad using IoT-based engineering examples, helping students map safeguards to threats. Use AI to analyse their own digital footprint and compare personal data exposure across platforms.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); CERT-In awareness materials (free); National Cyber Security Awareness Portal India (free)

UNIT II Cyber Threats, Cybercrime and Social Engineering

Core 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:

Use AI/LLM tools to explain the cyberattack lifecycle using a ransomware case study, with students mapping attacker actions to preventive controls. Analyse a sample spear-phishing email (for awareness only) to identify social engineering tactics.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); OWASP Top 10 (free public resource); Cisco Networking Academy (free)

UNIT III Personal Cyber Safety and Cybersecurity Technologies

Core 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). Safe use of AI tools and digital platforms.

AI Integration:

Use AI/LLM tools to explain MFA mechanisms (OTP, authenticator apps, hardware keys), and students set up Google Authenticator to understand TOTP conceptually. Use AI tools to understand public Wi-Fi risks and build a personal safe browsing checklist.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); Google Authenticator (free); Have I Been Pwned (free breach checker — https://haveibeenpwned.com)

UNIT IV Privacy, Cyber Ethics and Cyber Laws

Core Content:

Data protection and privacy concepts. Personal data and sensitive information. Cyber ethics and responsible digital behaviour. 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:

Use AI/LLM tools to explain key provisions of the IT Act 2000 and DPDP Act 2023, with students comparing data principal rights against GDPR. Use AI tools to understand the National Cyber Crime Reporting Portal and walk through a hypothetical complaint filing.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); cybercrime.gov.in (free); MeitY DPDP Act resources (free)

UNIT V	Cyber Hygiene and Secure Digital Practices

Core Content:

Cybersecurity risk awareness. Password management and Multi-Factor Authentication (consolidation). 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:

Use AI/LLM tools to guide students through a personal cyber hygiene audit covering password strength, MFA, and privacy settings, identifying key improvements. Build a personal Cyber Hygiene Action Plan with AI assistance and peer-review it using a structured rubric.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); NIST Cybersecurity Framework resources (free); CERT-In advisories (free); Cisco Networking Academy (free)

Course Outcomes (COs)

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

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L2	2	1	–	1	1	2	1	2	1	2	–	2	1	1
CO2	L2,L3	2	2	1	2	2	2	1	2	1	2	–	2	1	1

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO3	L3	2	2	2	2	3	3	1	2	1	2	1	2	2	2
CO4	L2	1	1	–	1	2	3	1	3	1	2	–	2	1	1
CO5	L3	2	2	2	2	2	3	1	3	1	2	1	3	1	2

Textbooks

1. Singh A., Introduction to Cyber Security, Universities Press / Orient BlackSwan, Latest Edition.
2. Godbole N. and Belapure S., Cyber Security: Understanding Cyber Crimes, Wiley India.

Reference Books

1. Brooks C.J. et al., Cybersecurity Essentials, Wiley.
2. Stallings W. and Brown L., Computer Security: Principles and Practice, Pearson.
3. Whitman M.E. and Mattord H.J., Principles of Information Security, Cengage Learning.

Online Resources

- CERT-In (Indian CERT) – https://www.cert-in.org.in
- National Cyber Crime Reporting Portal – https://cybercrime.gov.in
- NIST Cybersecurity Framework – https://www.nist.gov/cyberframework
- OWASP Foundation – https://owasp.org

I Year II Sem

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

Prerequisites: Basic English (Intermediate level)

Course Objectives

- Develop strong foundations in English grammar, vocabulary, phonetics, and communication models essential for professional contexts.
- Develop listening, speaking, reading, and writing skills using structured activities and AI-powered tools.
- Enable students to compose professional emails, reports, and presentations using AI writing assistants.
- Develop critical analysis of communication styles, barriers, and discourse structures using AI tools.
- Cultivate responsible AI usage for language learning — verifying AI outputs, maintaining academic integrity, and using AI as an aid not a substitute.

Course Content

UNIT I Foundations of English Communication

Core Content:

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.

AI Integration:

Grammar explanation, error identification, and instant Q&A on language rules, Real-time grammar, spelling, and punctuation correction with detailed explanations, Phonetics practice – voice recording for pronunciation and intonation feedback, Word-of-the-day, etymology, and interactive vocabulary quizzes.

Free/Open-Source AI Tools:

Grammarly (free); ChatGPT / Gemini (free tiers); Google Speech-to-Text (free); Merriam-Webster Vocabulary Builder (free)

UNIT II Listening and Speaking Skills

Core Content:

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.

AI Integration:

AI-powered presentation design – structured, visual, professional slides, Fluency analysis, filler-word detection, pacing, and confidence scoring, Auto-transcription and keyword analysis of spoken communication, Visual communication design – infographics, posters, and slide aesthetics

Free/Open-Source AI Tools:

Gamma.app (free); Orai AI Speech Coach (free); Otter.ai (free tier); Canva AI (free)

UNIT III Reading Comprehension & Writing Skills

Core Content:

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 and blogs.

AI Integration:

AI paraphrasing tool – rewrite passages at varying complexity levels, Readability scoring, passive voice detection, and sentence simplification, Essay outline generation, argument structuring, and content brainstorming, Vocabulary simplification for improving comprehension of complex texts

Free/Open-Source AI Tools:

QuillBot (free); Hemingway Editor (free online); ChatGPT (free); Rewordify (free)

UNIT IV Professional & Business Communication

Core Content:

Email Writing and Letter Writing: formal and informal. Short Report Writing: structure, types (formal/informal), and format. Note-Making and Summarization Techniques. Resume / CV Writing and LinkedIn Profile Optimization.

AI Integration:

Draft and refine professional emails, cover letters, reports, and proposals, AI-powered resume builder with ATS-optimized templates, Report structuring, agenda writing, and knowledge management, Tone detection, formality check, and professional register evaluation

Free/Open-Source AI Tools:

ChatGPT (free); Kickresume (free tier); Grammarly (free); Notion AI (free tier)

UNIT V Use of AI in Language Learning

Core Content:

Artificial Intelligence in Communication: meaning, basics, benefits and limitations of AI in human interaction. AI chatbots and virtual assistants in education: grammar correction, vocabulary building, AI-based speaking and pronunciation tools, writing assistants, e-portfolio creation. AI in Global Communication: automated translation, summarizing large texts, finding information in different languages. Digital Communication and AI Ethics: responsible use of AI in education, plagiarism and academic honesty, fake information and AI-generated content, privacy, bias, and ethical communication online.

AI Integration:

Plagiarism detection, paraphrase checker, and academic integrity analysis, High-accuracy translation supporting intercultural communication understanding, AI literacy modules – understanding AI bias, ethics, and responsible usage, E-portfolio creation platform for showcasing communication artifacts

Free/Open-Source AI Tools:

Copyleaks / Turnitin (institutional access); DeepL Translator (free); Wix (free); Adobe Portfolio (free for students); ChatGPT / Gemini (free tiers)

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Identify grammatical structures, phonetic patterns, and professional vocabulary, and recall communication frameworks using AI-assisted tools.	L1 (Remember) L2 (Understand)
CO2	Comprehend and interpret written and spoken communication models, and summarize content using AI paraphrasing and summarization tools.	L2 (Understand)
CO3	Compose professional emails, reports, and presentations using AI writing assistants such as ChatGPT, Grammarly, and Hemingway Editor.	L3 (Apply)
CO4	Analyse discourse structures, identify communication barriers, and use AI-driven sentiment analysis tools to evaluate message effectiveness.	L4 (Analyze)
CO5	Evaluate communication samples using AI grammar checkers, readability scorers, and peer-evaluation rubrics aligned to professional standards.	L5 (Evaluate)
CO6	Design and deliver professional communication artifacts — presentations, e-portfolios, and research reports — integrating AI tools for structure, style, and refinement.	L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	1	–	–	1	–	1	–	1	1	3	1	2	–	–
CO2	L2	1	–	–	1	1	2	1	1	2	3	1	2	–	–
CO3	L3	–	–	1	–	3	2	1	2	3	3	2	2	–	1
CO4	L4	–	1	1	2	2	2	1	2	2	3	2	2	–	1
CO5	L5	–	1	1	2	3	2	2	3	2	3	2	3	–	1
CO6	L6	–	1	2	2	3	3	2	3	3	3	3	3	–	2

Textbooks

1. Raman M. and Sharma S., Technical Communication: Principles and Practice, Oxford University Press, 4th Edition, 2022.
2. Shetty J. and Pareek R., Communicative English for Engineers and Professionals, Wiley India, 3rd Edition, 2021.

Reference Books

1. Rizvi M.A., Effective Technical Communication, Tata McGraw-Hill, 2nd Edition, 2018.
2. Murphy H.A., Effective Business Communication, McGraw-Hill Education, 7th Edition, 2019.
3. Wren P.C. and Martin H., High School English Grammar and Composition, S. Chand, Revised Edition, 2019.

Web Resources & NPTEL Links

- British Council – Learn English – https://learnenglish.britishcouncil.org
- NPTEL – Communication Skills – https://nptel.ac.in/courses/109104101
- Purdue OWL (Online Writing Lab) – https://owl.purdue.edu
- BBC Learning English – https://www.bbc.co.uk/learningenglish

I Year – II Semester

Course Code	ENGINEERING CHEMISTRY FOR ELECTRONICS AND COMMUNICATION SYSTEMS	L	T	Pr	P	Credits
26AHS00204T		3	0	0	0	3

Prerequisites: Basic Chemistry (Intermediate), Physics for Communication Systems

Course Objectives

- Understand quantum chemistry concepts relevant to electronic systems.
- Study spectroscopy techniques used in material and signal analysis.
- Analyse semiconductor and electronic material chemistry.
- Explore electrochemical energy storage for communication systems.
- Apply AI/ML tools for materials analysis, device reliability, and energy optimization.

Course Content

UNIT I Quantum Chemistry

Core Content:

Fundamentals of quantum chemistry — black body radiation, photoelectric effect, wave-particle duality, de-Broglie wave equation, Heisenberg's uncertainty principle. Fundamentals of quantum mechanics, Schrödinger wave equation, significance of Ψ and Ψ^2. Quantum numbers. Particle in a one-dimensional box. Molecular orbital theory — bonding in homo and hetero nuclear 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. Visualization of molecular orbitals and energy levels. Introduction to quantum computing models.

Free/Open-Source AI Tools:

Python (NumPy, SciPy), Google Colab

UNIT II Spectroscopy

Core Content:

Introduction, types of energy in molecules, types of spectra, representation and general features of absorption spectra. Microwave spectroscopy — introduction, principle, instrumentation and applications. UV-Visible spectroscopy — chromophore, auxochrome, bathochromic shift, hypsochromic shift, Beer-Lambert law, principle, instrumentation and applications. Infrared (IR) spectroscopy — introduction, principle, instrumentation and applications.

AI Integration:

AI-based spectral analysis and pattern recognition. Automated identification of compounds using machine learning models.

Free/Open-Source AI Tools:

Python (Pandas, scikit-learn), spectroscopy datasets, Google Colab

UNIT III Chemistry of Electronic Materials

Core Content:

Atomic structure and chemical bonding relevant to semiconductors. Band theory: valence band, conduction band, band gap energy. Intrinsic and extrinsic semiconductors: silicon, germanium, gallium arsenide. Doping chemistry: n-type (phosphorus, arsenic) and p-type (boron, aluminum) dopants. Chemical vapor 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 and fabrication optimization. Data-driven defect analysis in semiconductor wafers.

Free/Open-Source AI Tools:

TensorFlow, Python, Scilab

UNIT IV Electrochemistry and Energy Storage

Core Content:

Fundamentals of electrochemistry: redox reactions, electrode potential, Nernst equation. Galvanic cells and electrolytic cells. Energy storage: batteries — primary batteries (zinc-air, sodium-air), secondary batteries (lead-acid, NiCd, NiMH), lithium-ion batteries ($LiFePO_4$). Supercapacitors: EDLC, pseudo capacitors for energy harvesting. Fuel cells: PEMFC, DMFC for backup power. Energy harvesting: solar cells (dye-sensitized, perovskite), thermoelectric generators. Battery management: charging algorithms, state of charge, thermal management.

AI Integration:

AI models for battery health monitoring and energy optimization in IoT devices. Predictive analytics for battery life and performance.

Free/Open-Source AI Tools:

Python (Pandas, scikit-learn), Google Colab

UNIT V Corrosion Chemistry and Protection

Core Content:

Corrosion: definition, 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 (SiO_2, Si_3N_4, Al_2O_3). Corrosion of printed circuit boards: copper oxidation, tin whiskers. Corrosion in solder joints and interconnects. Protective techniques: proper design, pure metal, metal alloys, inhibitors. Cathodic protection. Electroplating (nickel) and electroless plating (copper).

AI Integration:

Computer vision for PCB defect detection. AI-based prediction of corrosion failure in electronic components.

Free/Open-Source AI Tools:

OpenCV, TensorFlow, Python

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply quantum chemistry principles to analyse electronic systems and predict molecular orbital properties.	L3 (Apply) L4 (Analyze)
CO2	Analyse spectroscopic techniques for material characterization and interpret spectral data for molecular identification.	L3 (Apply) L4 (Analyze)
CO3	Evaluate semiconductor materials and chemical fabrication processes relevant to electronic device manufacturing.	L3 (Apply) L4 (Analyze)
CO4	Analyse energy storage systems and electrochemical principles for communication device power management.	L3 (Apply) L4 (Analyze)
CO5	Design corrosion prevention and reliability strategies for electronic components using AI-assisted analysis.	L5 (Evaluate) L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3,L4	3	3	2	3	2	1	1	1	1	2	1	2	3	2
CO2	L3,L4	3	3	3	2	3	1	1	1	1	2	1	2	3	3
CO3	L3,L4	3	3	3	2	3	1	2	1	2	2	1	2	3	3
CO4	L3,L4	2	3	2	3	3	1	2	1	2	2	1	2	3	3
CO5	L5,L6	3	2	3	3	3	2	2	2	2	3	2	3	3	3

Textbooks

1. Jain P.C. and Jain M., Engineering Chemistry, 17th Edition, Dhanpat Rai Publishing Company, 2020.
2. Agarwal S., Engineering Chemistry, 5th Edition, Cambridge University Press, 2026.

Reference Books

1. Streetman B.G. and Banerjee S.K., Solid State Electronic Devices, 7th Edition, Pearson Education, 2015.
2. Sze S.M. and Ng K.K., Physics of Semiconductor Devices, 3rd Edition, John Wiley & Sons, 2006.
3. Dara S.S. and Umare S.S., Engineering Chemistry, 15th Edition, S. Chand Publishing, 2018.

Web Resources & NPTEL Links

- IEEE Electron Devices Society – https://eds.ieee.org
- SEMI – Semiconductor Manufacturing Standards – https://www.semi.org
- NPTEL Engineering Chemistry – https://nptel.ac.in

I Year II Sem

Course Code	DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS	L	T	Pr	P	Credits
26AHS00203T	Common to All Branches	3	0	2	0	4

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.
- 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 Content

UNIT I	First-Order Ordinary Differential Equations

Core Content:

Classification of ODEs by order, degree, and linearity; linear first-order ODEs and integrating factor method; Bernoulli's equation; exact equations and integrating factors; orthogonal trajectories; Newton's law of cooling 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 with initial conditions.

Free/Open-Source AI Tools:

Python (SymPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; Wolfram Alpha free tier for solution verification; GNU Octave for ODE routines.

UNIT II	Higher-Order Linear Differential Equations

Core Content:

Superposition principle; Wronskian and linear independence; complementary function for constant-coefficient ODEs (distinct real, repeated, and complex roots); particular integral by operator method; variation of parameters; simultaneous linear ODEs; Cauchy's (Euler) equation; Legendre's equation; applications to LCR circuits, simple harmonic motion, damped and forced oscillations; mass-spring-dashpot systems.

AI Integration:

Python SciPy signal.lsim for simulation of underdamped, critically damped, and overdamped responses; resonance amplitude curve generation; AI-assisted Wronskian computation and verification using SymPy; using AI tools to explain transient vs. steady-state response; resonance and Q-factorinterpretation.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, SymPy, Matplotlib) on Google Colab; Jupyter Notebook; GNU Octave / MATLAB (optional) for LCR and mechanical system simulation; WolframAlpha free tier.

UNIT III	Partial Differential Equations

Core Content:

Introduction and classification of PDEs; formation by elimination of arbitrary constants and functions; first-order linear PDEs and Lagrange's method; homogeneous linear PDEs with constant coefficients (CF and PI); standard forms of nonlinear first-order PDEs; second-order PDE applications — wave equation, heat equation, and Laplace's equation; classification as hyperbolic, parabolic, and elliptic.

AI Integration:

Python SymPy for PDE formation and symbolic verification; 3D surface plot visualization of PDE solution families using Matplotlib; SciPy finite-difference simulation of the 1D heat equation with animation; AI-assisted PDE classification using the discriminant; prompting AI to classify mixed-derivative PDEs and student verification of AI reasoning.

Free/Open-Source AI Tools:

Python (SymPy, NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; WolframAlpha free tier for PDE classification verification; Scilab.

UNIT IV	Vector Differentiation

Core Content:

Scalar and vector fields; del (∇) operator; gradient and directional derivative; unit normal to surfaces; angle between surfaces; divergence and physical interpretation; curl and physical interpretation; vector identities; conservative fields and scalar potentials; Laplacian; engineering applications in electromagnetic and fluid mechanics contexts.

AI Integration:

Python NumPy and Matplotlib for 2D and 3D gradient, divergence, and curl field visualization (quiver plots, level-surface overlays); SymPy for symbolic verification of vector identities; using AI tools to interpret Maxwell's equations in terms of divergence and curl operators; AI-assisted explanation of irrotational and solenoidal field classification.

Free/Open-Source AI Tools:

Python (NumPy, SymPy, Matplotlib, mpl_toolkits) on Google Colab; Jupyter Notebook; SciPy for numerical gradient computation; GNU Octave.

UNIT V	Vector Integration and Integral Theorems

Core Content:

Line integrals over scalar and vector fields; work done and path independence; surface integrals and flux; Green's theorem 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.

AI Integration:

Python SciPy integrate for numerical computation of line and surface integrals; 3D flux visualization with colour-coded normal flux using Matplotlib; numerical verification of Green's and Stokes' theorems; AI-assisted physical interpretation of the Divergence theorem in terms of source density; prompting AI to explain connections to Gauss's law and student evaluation of AI explanations.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; SymPy for symbolic integral verification; WolframAlpha free tier; 3D plotting tools.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Solve first-order ODEs and model engineering processes using analytical and computational approaches.	L3 (Apply)
CO2	Analyse 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.	L4 (Analyze)
CO5	Apply Green's, Stokes', and Divergence theorems to compute work, circulation, and flux and evaluate engineering field behaviour.	L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	2	1	–	–	–	1	2	–	–	2	1
CO2	L4	3	3	3	2	2	–	–	–	1	2	–	–	2	1
CO3	L3	3	2	2	2	1	–	–	–	–	2	–	–	2	1
CO4	L4	3	3	3	3	2	–	–	–	1	2	–	–	2	2
CO5	L5	3	3	2	3	3	–	1	1	2	2	1	–	2	2

Textbooks

1. Kreyszig E., Advanced Engineering Mathematics, 10th Edition, Wiley, 2018.
2. Grewal B.S., Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2017.
3. Thomas G.B., Weir M.D., Hass J., Thomas' Calculus, 14th Edition, Pearson, 2018.

Reference Books

1. Zill D.G. and Wright W.S., Advanced Engineering Mathematics, 6th Edition, Jones & Bartlett, 2018.
2. Sneddon I.N., Elements of Partial Differential Equations, Dover Publications, 2006.
3. Arfken G.B. et al., Mathematical Methods for Physicists, 7th Edition, Academic Press, 2012.

Web Resources & NPTEL Links

- NPTEL: ODEs, PDEs, and Vector Calculus – https://nptel.ac.in
- MIT OpenCourseWare – Differential Equations (18.03) and Multivariable Calculus (18.02): https://ocw.mit.edu
- Python SymPy Documentation: https://docs.sympy.org
- SciPy Documentation: https://docs.scipy.org

I Year II Sem

NETWORK ANALYSIS

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEE02201T	I Year II Sem	2	0	2	0	3

Prerequisites: Basic Electrical Engineering, Engineering Mathematics

Course Objectives

- To understand types of circuit elements, source transformations, and methods of mesh and nodal analysis for resistive networks.
- To apply fundamental network theorems for circuit simplification and problem solving.
- To analyse transient responses of RL, RC, and RLC circuits and solve them using differential equations and Laplace transform techniques.
- To perform steady-state AC circuit analysis using impedance, phasor notation, and resonance concepts.
- To characterise two-port networks using parameters and apply them to cascaded and matched network configurations.

Course Content

UNIT I Circuit Analysis of Resistive Networks & Network Theorems

Core Content:

Types of circuit components, types of sources and source transformations, mesh analysis and nodal analysis, problem solving with resistances only including dependent sources. Principle of duality with examples. Network Theorems: Thevenin's, Norton's, Reciprocity, Superposition, and Maximum Power Transfer theorems — problem solving using dependent sources.

AI Integration:

Use AI/LLM tools to explain mesh and nodal analysis with step-by-step worked examples; solve resistive circuits using KVL/KCL and prints branch currents; prepare a mesh vs. nodal comparison table and validate it using AI tools.

Free/Open-Source AI Tools:

Python (NumPy, SymPy) on Google Colab; PartSim, Falstad Circuit Simulator; ChatGPT / Gemini; NPTEL Network Analysis lectures.

UNIT II Transients & Laplace Transform

Core Content:

First-order differential equations, definition of time constants, R-L circuit and R-C circuit with DC excitation, evaluating initial conditions procedure. Second-order differential equations — homogeneous and non-homogeneous — problem solving using R-L-C elements with DC and AC excitation. Laplace transform: introduction, basic theorems, problem solving using Laplace transform, and partial fraction expansion.

AI Integration:

Use AI/LLM tools to explain the time constant of RL/RC circuits and sketch voltage/current evolution; plot step responses and shows how τ shifts with component values; demonstrate partial fraction expansion and Laplace-based RLC solutions, check answers against initial and final conditions.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; Scilab; Falstad Circuit Simulator; ChatGPT / Gemini; NPTEL video lectures.

UNIT III Steady State Analysis of AC Circuits

Core Content:

Impedance concept, phase angle, series R-L, R-C, and R-L-C circuits — problem solving. Complex impedance and phasor notation for R-L, R-C, and R-L-C circuits — problem solving using mesh and nodal analysis. Star-Delta conversion.

AI Integration:

Use AI/LLM tools to explain impedance, admittance, and phasor notation; solve a series R-L-C circuit for Z, I, and phase angle. plot impedance vs. frequency to identify resonant frequency; demonstrate Star-Delta conversion and compare series vs. parallel R-L-C behaviour.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; AC phasor calculators online; ChatGPT / Gemini; NPTEL AC circuits lectures.

UNIT IV Resonance & Coupled Circuits

Core Content:

Resonance: introduction, definition of Q-factor, series resonance, bandwidth of series resonance, parallel resonance, general case with resistance in both branches. Coupled Circuits: self-inductance, mutual inductance, coefficient of coupling, analysis of coupled circuits, natural current, dot rule of coupled circuits — problem solving.

AI Integration:

Use AI/LLM tools to explain resonant frequency, Q-factor, and bandwidth; verify f_0 and Q and confirm bandwidth = f_0/Q, plot the frequency response; mark the −3 dB points, read bandwidth, and compare with analytical results; apply the dot rule to a coupled circuit for polarity verification.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini; NPTEL resonance and coupled circuits lectures.

UNIT V Two-Port Networks

Core Content:

Relationship of two-port networks, Z-parameters, Y-parameters, Transmission (ABCD) parameters, h-parameters, relationships between parameter sets. Series and parallel connection of two-port networks, cascading of two-port networks — problem solving using dependent sources. Insertion loss, attenuators and pads, impedance matching networks.

AI Integration:

Use AI/LLM tools to explain Z, Y, h, and ABCD parameters and demonstrate Z-to-Y conversion via matrix inversion; compute and validate Z/Y parameters; explain ABCD cascading and verify output voltage for two cascaded T-networks with a given load.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; two-port network calculators online; ChatGPT / Gemini; NPTEL two-port network lectures; Scilab.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Analyse resistive networks using mesh analysis, nodal analysis, source transformations, and network theorems (Thevenin's, Norton's, Superposition, Maximum Power Transfer) to determine voltages, currents, and power in circuits with dependent sources.	L3 (Apply) L4 (Analyse)
CO2	Apply Laplace transform techniques and initial condition procedures to solve transient responses of first-order RL and RC circuits and second-order RLC circuits under DC and AC excitation.	L3 (Apply) L4 (Analyse)
CO3	Perform steady-state AC circuit analysis using phasor notation, complex impedance, and Star-Delta conversion for series and parallel R-L-C circuits using mesh and nodal methods.	L3 (Apply)
CO4	Evaluate series and parallel resonance characteristics including resonant frequency, Q-factor, and bandwidth, and analyse magnetically coupled circuits using self-inductance, mutual inductance, and the dot rule.	L4 (Analyse) L5 (Evaluate)
CO5	Derive and apply Z, Y, h, and ABCD two-port network parameters for the analysis, cascading, and design of attenuators, pads, and impedance matching networks.	L4 (Analyse) L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	Bloom's Level	PO1 Eng. Know.	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 PrjMgmt	PO12 Lifelong Learn.	PSO1 ECE Core	PSO2 Profnl Dev.
CO1	L3, L4	3	3	2	2	2	-	-	-	1	2	-	1	3	2
CO2	L3, L4	3	3	2	2	2	-	-	-	-	1	-	2	3	2
CO3	L3	3	3	2	1	2	-	-	-	-	1	-	1	3	1
CO4	L4, L5	3	3	2	2	2	-	-	-	1	2	-	2	3	2
CO5	L4, L5	3	3	3	2	2	-	-	-	1	2	1	2	3	3

Textbooks

1. Van Valkenburg M.E., Network Analysis, Revised 3rd Edition, Prentice Hall of India, 2019.
2. Hayt W.H., Kemmerly J., and Durbin S.M., Engineering Circuit Analysis, 9th Edition, McGraw-Hill, 2020.

Reference Books

1. Roy Choudhury D., Networks and Systems, New Age International Publications, 2013.
2. Edminister J. and Nahvi M., Electric Circuits, Schaum's Outline Series, 7th Edition, Tata McGraw Hill, 2017.
3. Ryder J.D., Network Lines and Fields, 2nd Edition, PHI.
4. Alexander C.K. and Sadiku M.N.O., Fundamentals of Electric Circuits, McGraw-Hill Education.

Web Resources & NPTEL Links

- NPTEL: Network Analysis – https://nptel.ac.in/courses/108104002
- MIT OpenCourseWare – Circuits and Electronics: https://ocw.mit.edu/courses/6-002
- Falstad Circuit Simulator (free online): https://www.falstad.com/circuit
- Python / Google Colab: https://colab.research.google.com
- PartSim Free Circuit Simulator: https://www.partsim.com
- Scilab (free MATLAB alternative): https://www.scilab.org

Recommended Free & Open-Source AI/Computational Tools

- Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for circuit simulation and Laplace transform analysis
- Scilab (free, open-source) – For matrix-based circuit and two-port network computations
- Falstad Circuit Simulator (free, open-source) – For interactive circuit visualisation and transient analysis
- PartSim (free online) – SPICE-based circuit simulation for verification

- ChatGPT / Gemini (free tiers) – AI/LLM tools for concept explanation and step-by-step problem walkthrough
- NPTEL Video Lectures – Free network analysis and circuit theory courses

I Year II Sem

Course Code	ENGINEERING GRAPHICS	L	T	Pr	P	Credits
26AME03201T	Common to All Branches	1	0	4	0	3

Prerequisites: Basic Geometry, Fundamental Python Programming

Course Objectives

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

Course Content

UNIT I Introduction to Engineering Graphics, Curves, and Scales

Core Content:

Introduction: principles of Engineering Graphics, BIS conventions, and lettering. Curves: construction of polygons, conic sections (ellipse, parabola, and hyperbola using eccentricity/general methods), cycloidal curves and involutes. Scales: plain, diagonal, and Vernier scales.

AI Integration:

AI prompt engineering for generating geometric profiles; algorithmic generation of conic curves using mathematical code blocks; parsing hand-drawn geometric sketches into clean vector shapes using Computer Vision (CV) edge detection (Canny Edge/Hough Transform).

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib, OpenCV) on Google Colab (free); AutoCAD student version (free); ChatGPT / Gemini for geometric concept explanation

UNIT II Orthographic Projections of Points, Lines and Planes

Core Content:

Principles of orthographic projections, projection of points in all four quadrants. Projection of straight lines inclined to one or both reference planes, traces of lines, determination of true lengths and true inclinations. Projections of Planes: regular planes perpendicular to both reference planes, parallel to one reference plane and inclined to the other; planes inclined to both reference planes.

AI Integration:

Machine learning pipelines for 3D coordinate estimation from 2D views; multi-view structural tracking; training neural models to predict true line length and orientation angles based solely on coordinate projections.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); AutoCAD student version (free); AI tools: ChatGPT / Gemini

UNIT III	Projection of Regular Solids

Core Content:

Types of solids: polyhedra and solids of revolution. Projections of solids in simple positions: axis perpendicular to horizontal plane, axis perpendicular to vertical plane, axis parallel to both reference planes. Projection of solids with axis inclined to one reference plane and parallel to another.

AI Integration:

Object recognition models (e.g., YOLO / MobileNet) trained to identify standard solid geometries; automated multiview projection generation from 3D CAD meshes using programmatic render layers.

Free/Open-Source AI Tools:

Python (scikit-learn, TensorFlow, OpenCV) on Google Colab (free); AutoCAD student version (free); FreeCAD (free open-source CAD)

UNIT IV	Sections of Solids and Development of Surfaces

Core Content:

Sections of Solids: perpendicular and inclined section planes, sectional views and true shape of section, sections of solids in simple positions only. Development of Surfaces: methods of development — parallel line development and radial line development; development of a cube, prism, cylinder, pyramid, and cone.

AI Integration:

Generative Adversarial Networks (GANs) for generating flat-pattern configurations; algorithmic optimization of surface layouts to minimize sheet manufacturing scrap; predictive toolpath generation using physics-guided heuristic networks.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib, DEAP library) on Google Colab (free); AutoCAD student version (free); AI tools: ChatGPT / Gemini

UNIT V Isometric Projections, Conversions, and Computer Graphics

Core Content:

Principles of isometric projection, isometric scale, isometric views/projections of planes, simple and compound solids. Conversion of isometric views to orthographic views and vice-versa. Computer Graphics: creating 2D and 3D drawings of objects including PCB and transformations using AutoCAD (not for end examination).

AI Integration:

2D-to-3D depth reconstruction using Neural Radiance Fields (NeRF) or specialized generative models; auto-generation of 3D CAD models from simple multi-view engineering layout sketches.

Free/Open-Source AI Tools:

Python (NumPy, Open3D, Matplotlib) on Google Colab (free); AutoCAD student version (free); FreeCAD (free); AI tools: ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Construct engineering curves and scales using traditional drafting geometry validated by automated algorithmic paths.	L3 (Apply)
CO2	Solve multi-planar projection instances of linear points, segments, and standard regular planar silhouettes.	L3 (Apply) 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) L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	2	1	–	–	–	–	–	–	2	–	1	2	1
CO2	L3,L4	3	3	2	–	–	–	–	–	–	2	–	1	2	1
CO3	L4	3	3	2	1	–	–	–	–	–	2	–	1	2	2
CO4	L4	3	3	3	1	2	–	–	–	–	2	–	2	3	2
CO5	L5,L6	3	2	2	–	3	–	–	–	1	3	–	2	3	2

Textbooks

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

Reference Books

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

Web Resources & Video Links

- NPTEL Engineering Graphical Design Course by IIT Kharagpur
- OpenCV Official Python Bootcamps and Tutorials
- AutoCAD Student Version – https://www.autodesk.com/education
- FreeCAD (free open-source CAD) – https://www.freecad.org

ENGINEERING WORKSHOP

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AME03201P	I Year II Sem	0	0	0	3	**1.5**	Common to All Branches

Prerequisites: Basic Computing Elements

Course Objectives

- Create awareness on safety precautions required at the workplace.
- Master fundamental hand tools and traditional manufacturing craftsmanship.
- Practice manufacturing of components using workshop trades including fitting, sheet metal, carpentry, foundry, and welding.
- Impart basic electrical engineering knowledge for house wiring practice.
- Demonstrate fundamentals of foundry and welding.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Level
CO1	Recognise the importance of safety precautions while working in the workshop.	L1 (Remember)
CO2	Identify different measuring, marking, and cutting tools used in the workshop and their operational capabilities.	L2 (Understand)
CO3	Fabricate precision wood and metal fitting joints within standard specifications.	L3 (Apply)
CO4	Understand wire standards, safe residential switches, lighting circuits, and basic appliances.	L2 (Understand)
CO5	Identify different tools used in foundry and welding sections.	L2 (Understand)

List of Experiments / Trades

1. Safety Practices & Precautions

Demonstrate the necessity of workshop hazards, various Personal Protective Equipment (PPE) and emergency protocols.

- Types of workshop hazards: mechanical, electrical, chemical, and fire hazards.
- Personal Protective Equipment (PPE): safety goggles, gloves, helmets, ear muffs, aprons.

- Emergency protocols: fire extinguisher operation, first-aid procedures, emergency exits.
- Housekeeping and safe disposal of waste materials.

AI Integration: *Computer vision for real-time safety compliance monitoring — detecting PPE violations and unsafe zones using camera feeds.*

2. Wood Working — Carpentry

Introduction to different types of woods, carpentry tools, and timber joints.

Exercises:

- (a) Half-Lap Joint
- (b) Mortise and Tenon Joint
- (c) Corner Dovetail Joint or Bridle Joint

3. Fitting

Familiarity with different types of tools used in fitting and perform the following fitting exercises.

Exercises:

- (a) V-Fit
- (b) Dovetail Fit
- (c) Semi-Circular Fit

4. Sheet Metal Working

Familiarity with different types of tools used in sheet metal working. Development of the following sheet metal jobs from GI sheets.

Exercises:

- (a) Tapered Tray
- (b) Conical Funnel
- (c) Elbow Pipe
- (d) Brazing

5. Electrical Wiring

Introduction to electrical tools, safety precautions, and wiring materials. Familiarity with different types of basic electrical circuits and make the following connections.

Exercises:

- (a) Parallel and Series circuits
- (b) Two-way Switch
- (c) Godown Lighting
- (d) Tube Light connection
- (e) Three-phase Motor connection

- (f) Soldering of wires

6. Plumbing

Introduction to plumbing tools, pipe fittings, and thread cutting. Making simple PVC pipe joints.

- Preparation of pipe joints with coupling for same-diameter pipes.
- Preparation of pipe joints with reducer for different-diameter pipes.

7. Welding Shop

Demonstration and practice on Arc Welding and Gas Welding.

- Identification and use of welding tools and equipment.
- Arc welding: electrode selection, current setting, striking an arc.
- Gas welding: oxy-acetylene setup, flame types, torch handling.
- Preparation of a simple Lap Joint / Butt Joint.

8. Foundry Trade

Demonstration and practice on moulding tools and processes.

- Identification of foundry tools: hand rammer, sprue pin, trowel, lifter, draw spike.
- Properties of moulding sand: green sand composition and preparation.
- Preparation of Green Sand Moulds for a simple pattern.

9. Introduction to 3D Printing

Demonstrate the core concepts of 3D printing — hardware, software, digital file formats, and 3D printing materials.

- Overview of 3D printing technologies: FDM, SLA, SLS.
- Hardware components: extruder, print bed, frame, stepper motors.
- Software: slicing software (Cura / PrusaSlicer), STL file preparation.
- 3D printing materials: PLA, ABS, PETG — properties and applications.
- Familiarise with 3D printing process of complex objects including support structures and post-processing.

AI Integration: *AI-driven slicing optimisation — using machine learning to auto-select infill patterns, support geometry, and print parameters for minimal material waste.*

10. Computer Components, Networking & Peripheral Installation

Disassembly and Reassembly:

- Disassemble and reassemble a desktop computer to understand the role of each component (CPU, motherboard, RAM, HDD/SSD, GPU, PSU, etc.).

Hardware Troubleshooting:

- Simulate common hardware issues (RAM failure, CPU overheating) and identify procedures for resolving them.

Peripheral Installation:

- Install and configure peripheral devices such as printers, scanners, and external hard drives.

Network Setup and Configuration:

- Set up a small LAN network with routers, switches, and computers.
- Configure IP addresses, subnet masks, and basic network services.

11. Disassembly and Assembly of Mechanical Equipment

Familiarise with the disassembly and assembly process of the following equipment.

- Bicycle — frame, wheel bearings, chain drive, braking system.
- Gear Box — gear types, shaft arrangement, lubrication.
- Braking System — drum and disc brake components, hydraulic lines.

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

CO – PO Mapping

Scale: 1 = Low / Slight | 2 = Medium / Moderate | 3 = High / Substantial | – = No Correlation

CO / PO	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

Learning Resources

Text Books

S.No	Title	Author(s)	Publisher	Year
1	Workshop Technology Vol. I & II	S. K. Hajra Choudhury	Media Promoters & Publishers	—
2	A Course in Workshop	B. S. Raghuwanshi	Dhanpat Rai & Co.	2015 / 2017

S.No	Title	Author(s)	Publisher	Year
	Technology			
3	Hands-On Industrial Internet of Things	Giacomo Veneri & Antonio Capasso	Packt Publishing	—
4	Manufacturing Process	Felix W.	Independently Published	2019
5	Workshop Processes, Practices and Materials (5th Edn.)	Bruce J. Black	Routledge	2015
6	Wiring Estimating, Costing and Contracting	Soni P.M. & Upadhyay P.A.	Atul Prakashan	2021-22

Reference Books

S.No	Title	Author(s)	Publisher	Year
1	Manufacturing Technology (Foundry, Forming and Welding)	P. N. Rao	McGraw Hill	—
2	An Introduction to Predictive Maintenance	R. Keith Mobley	Butterworth-Heinemann	—

Video Links & Online Lectures

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

I Year II Sem

ENGINEERING CHEMISTRY FOR ELECTRONICS AND COMMUNICATION SYSTEMS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AHS00204P	I Year II Sem	0	0	0	3	1.5	

Prerequisites: Engineering Chemistry (concurrent theory course); Basic Sciences

Course Objectives

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

List of Experiments

Part A – Regular Experiments

1. Estimation of Ferrous Iron (Fe^{2+}) using Potassium Dichromate ($K_2Cr_2O_7$) – Redox Titrations
2. Determination of Strength of Acid in Lead Acid Battery
3. Determination of Corrosion Rate (Weight Loss Method)
4. Determination of Corrosion by Environment (Effect of pH, Salt Concentration, and Temperature)

Instrumental Methods of Chemical Analysis

5. Conductometric Titration of Strong Acid vs. Strong Base (HCl vs. NaOH)
6. Conductometric Titration of Weak Acid vs. Strong Base (CH_3COOH vs. NaOH)
7. Potentiometry (Redox Titration) – Estimation of Iron (Fe^{2+}) using Potassium Dichromate ($K_2Cr_2O_7$)
8. Preparation of a Polymer (Bakelite)
9. Spectrophotometry – Determination of Wavelength (λ) and Verification of Beer–Lambert Law
10. Identification of Functional Groups in Organic Compounds by IR Spectrophotometer

Part B – AI Integration Lab Exercises

- Titration Data Automation (Python / Google Colab): input burette readings from Experiments 1, 2, 5, 6, and 7; compute titre volumes, molarity, and strength; plot titration curves using Matplotlib; automate endpoint detection from conductance or potential vs. volume data.
- AI-Based Corrosion Prediction (scikit-learn): using a provided dataset of metal type, pH, temperature, salt concentration, and exposure time, train a Random Forest Regression model to predict corrosion rate; evaluate with R^2 and RMSE; compare predicted values with Experiment 3 and 4 results.
- Spectral Data Analysis (Python / NumPy): input absorbance vs. wavelength data from Experiment 9; plot absorbance spectrum; verify Beer–Lambert law by linear regression of absorbance vs. concentration; determine molar absorptivity (ε) and compare with standard values.

AI Integration	AI integration adds three exercises following the regular experiments: (i) Python automation of titration curve plotting and endpoint detection from raw burette data (Experiments 1, 2, 5, 6, 7) using Matplotlib/NumPy; (ii) scikit-learn Random Forest regression for predicting corrosion rate from environmental parameters, validated against Experiment 3 and 4 results, with R^2 and RMSE evaluation; (iii) Python spectral regression for Experiment 9 – fitting absorbance vs. concentration data, computing molar absorptivity, verifying Beer–Lambert law, and overlaying the theoretical linear relationship.

Free / Open-Source AI Tools: Python (NumPy, Matplotlib, scikit-learn) – Google Colab (free); LibreOffice Calc (free) – data recording; Virtual Chemistry Lab – VLabs IIT (free).

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Perform redox and acid-base titrations (Experiments 1, 2); determine iron (Fe^{2+}) concentration using $K_2Cr_2O_7$ and acid strength in battery electrolytes; compute molarity, normality, and percentage strength from titre data; interpret equivalence point from titration curves.	L3 (Apply)
CO2	Measure corrosion rates by weight loss and electrochemical methods (Experiments 3, 4); evaluate the effect of environmental variables (pH, temperature, NaCl concentration) on corrosion behaviour; compute corrosion rate in mdd or mpy units and interpret passivation trends.	L3–L4 (Apply, Analyse)

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO3	Perform conductometric and potentiometric titrations (Experiments 5, 6, 7); identify equivalence points from conductance vs. volume and potential vs. volume plots; explain the difference in conductance profiles between strong-acid/strong-base and weak-acid/strong-base systems; determine concentration of unknown samples.	L4 (Analyse)
CO4	Synthesise Bakelite (Experiment 8); determine the wavelength of maximum absorption and verify Beer–Lambert law using spectrophotometry (Experiment 9); identify functional groups in organic compounds from IR spectra (Experiment 10); correlate absorption peaks with molecular structure.	L4–L5 (Analyse, Evaluate)
CO5	Automate titration curve plotting and endpoint detection using Python (NumPy/Matplotlib); develop an AI-based corrosion rate predictor using scikit-learn Random Forest regression; perform spectral regression to verify Beer–Lambert law and compute molar absorptivity; evaluate model performance using R^2 and RMSE.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	State Faraday's law; define BHN; recall Beer–Lambert law; state Nernst equation; define equivalent weight
L2 – Understand	Explain tests, describe procedures	CO1–CO5	Explain why conductance dips at equivalence point; describe galvanic corrosion mechanism; explain IR spectral interpretation; describe Bakelite synthesis
L3 – Apply	Conduct tests, record, compute	CO1, CO2	Perform redox titration and compute Fe^{2+} concentration; measure corrosion rate under controlled conditions; calibrate spectrophotometer and record absorbance
L4 – Analyse	Compare, verify, interpret	CO2–CO4	Compare potentiometric and conductometric endpoint detection; analyse effect of pH on corrosion rate; verify Beer–Lambert linearity; interpret IR peaks for functional group identification
L5 –	Assess	CO4,	Evaluate % error in molar absorptivity; assess AI

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
Evaluate	accuracy, evaluate models	CO5	corrosion model R^2 and RMSE; evaluate experimental vs. theoretical titration curves
L6 – Create	Automate, develop, propose	CO5	Code Python titration curve automation; develop ML corrosion predictor; write Beer–Lambert regression script; propose a corrosion inhibitor formulation based on data trends

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 Com mun.	PO11 PrjMgmt	PO12 Lifelong Learn.	PSO1 ECE Core
CO1	L3	3	2	1	3	2	2	1	1	2	2	–	1	3
CO2	L3-L4	3	3	2	3	2	2	3	1	2	2	–	2	3
CO3	L4	3	3	2	3	3	1	–	1	2	2	–	2	3
CO4	L4-L5	3	3	2	3	3	1	1	1	2	2	–	2	3
CO5	L5-L6	2	3	3	3	3	1	1	1	2	3	1	3	2

Recommended Free & Open-Source AI / Computational Tools

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, Matplotlib, scikit-learn) on Google Colab (free) – titration curve automation, spectral regression, AI corrosion prediction
scikit-learn (free, open-source) – Random Forest regression for corrosion rate prediction from environmental parameters
Virtual Chemistry Lab – VLabs IIT (free): https://chem-iitb.vlabs.ac.in – simulation of titrations, electrochemical and spectroscopic experiments
LibreOffice Calc (free, open-source) – standardised data recording templates for all experiments
ChemDraw / MolView (free online) – molecular structure drawing and IR peak prediction for Experiment 10

Web Resources

- IEEE Electron Devices Society – https://eds.ieee.org
- SEMI (Semiconductor Equipment and Materials International) – https://www.semi.org
- Optica (OSA) – https://www.optica.org
- IEC (International Electrotechnical Commission) – https://www.iec.ch
- ECIA (Electronic Components Industry Association) – https://www.ecianow.org

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI Lab Exercises	3 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI exercises to be submitted as Google Colab .ipynb notebooks.

COMMUNICATIVE ENGLISH LAB – Common to All Branches

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AHS00201P	I Year II Sem	0	0	0	2	1	Common to All Branches

1. Course Objectives

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

2. Course Outcomes

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

Note: BTL = Bloom's Taxonomy Level (Revised) | 3 = High | 2 = Medium | 1 = Low correlation

3. Lab Syllabus – 12 Topics with AI Integration

S.No	Lab Topic	Lab Activities	AI Activity	AI Tools	COs
1	Phonetics & Pronunciation	Record & playback	Use AI speech-to-text to detect	ELSA Speak, Google	CO1, CO5

S.No	Lab Topic	Lab Activities	AI Activity	AI Tools	COs
	Practice	vowel/consonant articulation; IPA chart drill; Minimal pair exercises	mispronunciations; real-time phoneme feedback via ELSA Speak	Speech Recognition, Speechling	
2	Listening Comprehension & Note-Taking	Listen to TED Talks/podcasts; fill-in-the-blank exercises; summarise audio passages	Auto-generate transcripts; compare student notes with AI-generated summaries using Otter.ai	Otter.ai, YouTube Auto-captions, Whisper AI	CO2, CO5
3	Spoken English & Fluency Building	Impromptu speaking (1–2 min); picture description; story completion tasks	AI analyses speech for filler words, pace, and clarity; ChatGPT prompts for spontaneous speaking	Speeko, ChatGPT, Google Assistant	CO1, CO3, CO5
4	Group Discussion Techniques	Role-based GD on current topics; turn-taking practice; consensus building	AI evaluates argument structure and language; ChatGPT generates counter-arguments for practice	ChatGPT, Gemini, Mentimeter	CO2, CO3
5	Public Speaking & Presentations	Prepare and deliver 3-minute presentations; use of visual aids; Q&A handling	AI slide deck generation; real-time delivery feedback on pacing and confidence	Beautiful.ai, Gamma.app, Microsoft Copilot	CO3, CO5
6	Interview Skills & Mock Interviews	HR round simulation; STAR method answers; body language and eye contact	AI-powered mock interview with instant feedback on content, grammar, and tone	Yoodli, Interview Warmup by Google, ChatGPT	CO1, CO3, CO5
7	Professional Email & Business Writing	Draft formal emails, complaints, and requests; subject line formulation; netiquette	AI refines email drafts; tone & clarity check; compare student draft with AI-improved version	Grammarly, ChatGPT, Gemini	CO4, CO5
8	Resume &Cover	ATS-friendly	AI resume scanner;	Kickresume,	CO4,

S.No	Lab Topic	Lab Activities	AI Activity	AI Tools	COs
	Letter Writing	resume design; action verbs usage; tailoring cover letters to job descriptions	keyword optimisation suggestions; ChatGPT cover letter generation and critique	Resume.io, Jobscan, ChatGPT	CO5
9	Reading Comprehension & Critical Thinking	Read articles/editorials; identify main idea, inference, and tone; speed reading drills	AI generates comprehension questions; text summarisation and comparative analysis	Quillbot, Speechify, NotebookLM	CO2, CO4
10	Debate & Argumentation Skills	Structured debate (for/against); rebuttal techniques; logical fallacy identification	AI generates arguments on both sides; evaluate logical structure using ChatGPT	ChatGPT, Claude.ai, Kialo Edu	CO2, CO3
11	Report Writing & Technical Communication	Lab report, incident report, and project report formats; headings, data presentation	AI improves report coherence and passive/active voice usage; grammar and structure feedback	Grammarly Business, ChatGPT, Hemingway Editor	CO4, CO5
12	Etiquette, Netiquette & Personal Branding	Workplace conversation role play; LinkedIn profile creation; professional social media etiquette	AI reviews LinkedIn summaries; digital persona analysis; personal brand statement via ChatGPT	ChatGPT, Canva AI, LinkedIn AI features	CO3, CO4, CO5

4. Text Books

S.No	Text Books
1	Raman, Meenakshi & Sharma, Sangeeta. Technical Communication: Principles and Practice (3rd Ed.). Oxford University Press, 2022.
2	Krishnaswamy, N. & Krishnaswamy, L. The Story of English in India. Foundation Books, Cambridge University Press, 2021.

5. Reference Books

S.No	Reference
1	Swan, Michael. Practical English Usage (4th Ed.). Oxford University Press, 2016.
2	Murphy, Raymond. English Grammar in Use (5th Ed.). Cambridge University Press, 2019.
3	Rizvi, M. Ashraf. Effective Technical Communication (2nd Ed.). McGraw-Hill Education, 2018.
4	Turton, N.D. & Heaton, J.B. Longman Dictionary of Common Errors. Pearson Longman, 2017.
5	Lesikar, Raymond V. & Flatley, Marie E. Basic Business Communication (12th Ed.). McGraw-Hill, 2020.
6	Rutherford, Andrea J. Basic Communication Skills for Technology (3rd Ed.). Pearson, 2019.

6. Online Resources

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

7. 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)

8. 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

9. 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

10. CO – PO / PSO Mapping

Correlation Level: 3 = High | 2 = Medium | 1 = Low | – = 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

Mapping Legend

PO	Programme Outcome Description	Correlation with CEL	Strength
PO1	Engineering Knowledge / Disciplinary Knowledge	Grammar, vocabulary & technical language	2
PO4	Teamwork & Communication	GD, presentations, professional writing	3
PO5	Modern Tool Usage	AI tools for language learning	3
PO6	Professional Ethics	Workplace etiquette & netiquette	2
PO7	Life-long Learning	Self-paced AI-assisted improvement	2

AI Integration	AI integration is embedded across all 12 lab topics: (i) ELSA Speak and Whisper AI provide real-time phoneme and fluency feedback for pronunciation and listening labs; (ii) ChatGPT, Gemini, and Claude.ai drive argument generation, mock interview simulation, and writing refinement across GD, debate, email, resume, and report labs; (iii) Yoodli delivers AI-powered public speaking and interview coaching with pacing and confidence analytics; (iv) Grammarly and Hemingway Editor automate grammar, style, and readability correction for all written tasks; (v) Canva AI and Beautiful.ai assist in professional presentation design. Together these tools support self-paced improvement, personalised feedback, and continuous skill enhancement aligned with CO5.

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI Lab Exercises	Selected topics	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every session. AI exercise outputs (screenshots, logs, reflections) to be included in the lab record.

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

Prerequisites: None

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 Content

UNIT I Health, Wellness and Preventive Healthcare

Core Content:

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 and 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.

AI Integration:

Log BMI, steps and sleep for 2 weeks using Google Fit / Apple Health; generate AI health-insight reports and compare with WHO benchmarks.Use NutriAI / Eat Right Now app to design a 7-day balanced meal plan; compare output with ICMR dietary guidelines.

Free/Open-Source AI Tools:

Google Fit / Apple Health (free); Python (Pandas, Matplotlib) on Google Colab (free); ChatGPT / Gemini (free tiers); MyFitnessPal (free)

UNIT II Yoga, Pranayama, Meditation and Mind-Body Practice

Core Content:

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 and balance: Virabhadrasana I and 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.

AI Integration:

Use YogaNotch / KAIA Health app for real-time AI posture feedback; record before/after improvement videos.

Free/Open-Source AI Tools:

YogaNotch / KAIA Health (free tier); ChatGPT / Gemini (free tiers); Python (Pandas, Matplotlib) on Google Colab (free); Insight Timer (free guided meditation app)

UNIT III Physical Fitness, Sports Skills and Sports Technology

Core Content:

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. Sports technology — wearables, fitness trackers, GPS analytics, performance dashboards.

AI Integration:

Engage with Woebot / Wysa AI mental-health chatbot for one week; write a critical reflection on AI vs human counsellor effectiveness.Analyse a cricket/football dataset on Google Colab using Pandas; visualise player performance KPIs with Matplotlib / Seaborn.

Free/Open-Source AI Tools:

Python (Pandas, Matplotlib, Seaborn) on Google Colab (free); ChatGPT / Gemini (free tiers); Google Fit / Apple Health (free)

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Explain the fundamental concepts of health, wellness, fitness, and the dimensions of holistic well-being.	L1 (Remember) L2 (Understand)
CO2	Demonstrate correct yogic practices — asanas, pranayama, and meditation — for physical and mental health.	L3 (Apply)
CO3	Design a personal fitness plan integrating nutrition, exercise,	L3 (Apply) L4

CO	Course Outcome Statement	Bloom's Level
	stress management, and lifestyle modification.	(Analyze)
CO4	Understand rules, skills, and safety norms for a chosen sport and participate actively with sportsmanship.	L3 (Apply)
CO5	Evaluate personal health indicators and reflect on lifestyle choices using technology and evidence-based practices.	L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	2	–	–	–	–	3	2	–	–	–	–	–	–	–
CO2	L3	–	–	3	–	–	3	3	–	–	–	–	–	–	–
CO3	L3,L4	–	2	3	3	–	3	3	–	–	–	–	–	2	–
CO4	L3	–	–	3	–	–	2	2	–	3	–	–	–	–	–
CO5	L5	–	3	–	3	2	3	2	–	–	–	–	–	2	–

Textbooks

1. Mangal S.K. and Mangal S., Health and Physical Education, PHI Learning Pvt. Ltd., 2021.
2. Broad W.J., The Science of Yoga, Simon & Schuster, 2012.

Reference Books

1. Silva, Mira and Mehta S., Yoga: The Iyengar Way, Dorling Kindersley, 2019.
2. Clover J., Sports Medicine Essentials, Cengage Learning, 2020.
3. Bucher C.A., Foundations of Physical Education and Sport, McGraw-Hill Education, 2018.

Web Resources

- Ministry of AYUSH – https://main.ayush.gov.in
- National Health Portal – https://nhp.gov.in
- ICMR Dietary Guidelines – https://www.icmr.nic.in
- WHO Physical Activity Guidelines – https://www.who.int
- Khelo India Programme – https://kheloindia.gov.in

II Year I Sem

COMPLEX VARIABLES AND PROBABILITY THEORY

JNTUA R26 | II Year I Semester | Common to CE, ME, ECE & EEE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AHS00303T	II Year I Sem	3	0	0	0	3

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

Course Objectives

- Develop a thorough understanding of complex variable theory, analyticity, Cauchy-Riemann conditions, and harmonic functions as foundations for engineering analysis.
- Apply complex integration techniques, series expansions, and the residue theorem to evaluate contour integrals and real definite integrals.
- Formulate probabilistic models using axioms, random variable theory, and standard distributions, and compute correlation and regression measures.
- 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.

Course Content

UNIT I Complex Functions and Analyticity

Core Content:

Functions of a complex variable; limits and continuity; differentiability; analyticity; Cauchy-Riemann equations in Cartesian coordinates; Cauchy-Riemann equations in polar coordinates; harmonic functions; conjugate harmonic functions; Milne-Thomson method.

AI Integration:

Python-based visualization of complex functions and their mappings; symbolic verification of Cauchy–Riemann equations using SymPy; AI-assisted plotting of harmonic and conjugate harmonic surfaces; use of ChatGPT/Copilot to explain analyticity and the Milne–Thomsonconstruction.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib, SymPy) on Google Colab; Jupyter Notebook; WolframAlpha free tier for verification; GNU Octave.

UNIT II Complex Integration

Core Content:

Line integrals in the complex plane; Cauchy's integral theorem; Cauchy's integral formula; generalised Cauchy's integral formula; Liouville's theorem; maximum modulus theorem.

AI Integration:

Numerical evaluation of complex line integrals using SciPy; AI-assisted verification of Cauchy's integral theorem results; visualization of contour paths in the complex plane using Matplotlib; use of AI tools to interpret Liouville's theorem and maximum modulus principle.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib, SymPy) on Google Colab; Jupyter Notebook; WolframAlpha free tier; GNU Octave.

UNIT III Series Expansions and Residues

Core Content:

Taylor's series; Laurent's series; zeros and singularities: isolated, removable, poles, essential; residues; Cauchy's residue theorem; evaluation of real definite integrals using the residue theorem.

AI Integration:

SymPy-based Taylor and Laurent series generation for complex functions; AI-assisted classification of singularities and residue computation; Python visualization of convergence regions; numerical validation of definite integrals using SciPy and comparison with residue-based results.

Free/Open-Source AI Tools:

Python (SymPy, NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; WolframAlpha free tier for symbolic verification.

UNIT IV Probability and Random Variables

Core 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-based simulation of sample spaces and probability experiments; visualization of PMF, PDF, and CDF using Matplotlib; AI-assisted explanation of Bayes' theorem with real-world examples; NumPy/SciPy computation of expectation and variance for standard distributions.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy for symbolic probability computations; Jupyter Notebook; scikit-learn for exploratory data modelling.

UNIT V Probability Distributions

Core Content:

Binomial distribution: properties, mean and variance; Poisson distribution: properties, Poisson approximation to binomial; normal distribution: properties, standard normal curve, areas under normal curve; correlation coefficient; rank correlation; regression lines and regression coefficients.

AI Integration:

SciPy-based computation and visualization of binomial, Poisson, and normal distributions; AI-assisted plotting of standard normal curve and computation of area probabilities; Python-based correlation and regression analysis on engineering data sets; use of AI tools to interpret regression coefficients and correlation strength.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib, pandas) on Google Colab; scikit-learn for regression and correlation analysis; Jupyter Notebook; WolframAlpha free tier; GNU Octave.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Recall definitions of limits, continuity, and differentiability of complex functions and state the Cauchy-Riemann equations in Cartesian and polar forms.	L1 (Remember) L2 (Understand)
CO2	Apply Cauchy's integral theorem, Cauchy's integral formula, Liouville's theorem, and the maximum modulus theorem to evaluate complex line integrals.	L3 (Apply)
CO3	Analyse complex functions using Taylor's and Laurent's series, classify singularities, and evaluate real definite integrals by the residue theorem.	L4 (Analyze)
CO4	Understand the axioms and theorems of probability and explain the properties of discrete and continuous random variables including expectation and variance.	L2 (Understand)
CO5	Evaluate probabilities using binomial, Poisson, and normal distributions and interpret correlation and regression relationships in engineering data sets.	L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	3	1	–	–	–	–	–	–	–	–	–	–	2	1
CO2	L3	3	2	1	–	–	–	–	–	–	–	–	–	2	1

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO3	L4	3	3	2	2	1	–	–	–	–	–	–	–	2	2
CO4	L2	3	2	–	1	–	–	–	–	–	–	–	–	2	1
CO5	L5	3	3	1	2	1	1	–	–	1	1	–	–	2	2

Textbooks

1. Kreyszig E., Advanced Engineering Mathematics, 10th Edition, Wiley, 2011.
2. Walpole R.E. et al., Probability & Statistics for Engineers and Scientists, 9th Edition, Pearson, 2016.
3. Churchill R.V. and Brown J.W., Complex Variables and Applications, 9th Edition, McGraw-Hill, 2013.

Reference Books

1. Ross S.M., Introduction to Probability and Statistics for Engineers and Scientists, 5th Edition, Academic Press, 2014.
2. Grewal B.S., Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2021.
3. Gupta S.C. and Kapoor V.K., Fundamentals of Mathematical Statistics, 12th Edition, Sultan Chand & Sons, 2020.

Web Resources & NPTEL Links

- NPTEL: Complex Analysis – https://nptel.ac.in
- NPTEL: Probability and Statistics – https://nptel.ac.in
- MIT OpenCourseWare – Complex Variables: https://ocw.mit.edu
- Google Colab for Python: https://colab.research.google.com

II Year I Sem

BIOLOGY FOR ENGINEERS

JNTUA R26 | II Year II Semester | Common to All Branches

Course Code	Year/Sem	L	T	Pr	P	Credits
26AHS00304T	II Year II Sem	2	0	0	0	2

Prerequisites: Basic Sciences (Intermediate level)

Course Overview

This course introduces biological concepts through the lens of engineering and artificial intelligence, emphasising biomolecules, human systems, bio-inspired designs, and modern bioengineering trends, with practical case studies and AI-enabled problem-solving. It develops interdisciplinary thinking connecting biology, engineering, and computational tools.

Course Objectives

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

Course Content

UNIT I Foundations of Biology and AI

Core Content:

Cell as the basic unit of life. Carbohydrates, proteins, lipids, enzymes, vitamins, hormones — structure, function, and engineering relevance.

AI Integration:

AI-based visualization of biomolecules, Case studies on AI in drug discovery.

Free/Open-Source AI Tools:

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

UNIT II Biomolecules in Engineering Applications

Core Content:

Carbohydrates in bioplastics (PHA, PLA). Proteins in food engineering (whey protein, plant analogs). Lipids in biodiesel and detergents. AI-driven generative design for biomaterials.

AI Integration:

AI-driven generative design for biomaterials, Lipids in biodiesel & detergents.

Free/Open-Source AI Tools:

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

UNIT III Human Systems and Bio-Designs

Core Content:

Brain as CPU (EEG, robotic prosthetics). Eye as camera (optical corrections, lens materials). Heart as pump (ECG, stents). Kidney as filtration (dialysis systems).

AI Integration:

AI-based ECG anomaly detection, Robotic arm design inspired by neural signals, Use of AI imaging tools to detect cataracts, Simulate stent design using ANSYS.

Free/Open-Source AI Tools:

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

UNIT IV Bio-Inspired Materials and Mechanisms

Core Content:

Photosynthesis → photovoltaic cells. Bird flight → GPS and aircraft navigation. Lotus leaf → superhydrophobic surfaces. Kingfisher beak → bullet train design.

AI Integration:

Model solar energy conversion inspired by photosynthesis, AI simulation of lotus-leaf hydrophobicity, Kingfisher-inspired aerodynamic modelling, Use of AI to design self-cleaning surfaces.

Free/Open-Source AI Tools:

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

UNIT V Emerging Trends in Bioengineering

Core Content:

Muscular and skeletal scaffolds. 3D bioprinting of tissues (ear, bone, skin). AI in medical imaging (MRI, CT scans). AI in diagnostics and predictive healthcare.

AI Integration:

AI in medical imaging (MRI, CT scans), AI in diagnostics and predictive healthcare.

Free/Open-Source AI Tools:

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

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Explain the fundamentals of biology, biomolecules, and human systems relevant to engineering applications.	L2 (Understand)
CO2	Apply AI tools and computational techniques for biological data analysis, biomolecule visualisation, and healthcare applications.	L3 (Apply)
CO3	Analyse biomolecules and bio-inspired systems for solving engineering problems and developing sustainable technologies.	L4 (Analyze)
CO4	Design biomimetic and bioengineering solutions using interdisciplinary approaches and AI-enabled tools.	L5 (Evaluate) L6 (Create)
CO5	Evaluate emerging trends in bioengineering such as 3D bioprinting, regenerative medicine, and AI-based diagnostics for societal applications.	L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L2	3	2	1	1	1	–	–	–	–	–	–	2	2	1
CO2	L3	2	3	2	2	3	–	–	–	1	–	–	2	2	2
CO3	L4	2	3	3	2	2	2	3	–	–	–	–	2	2	2
CO4	L5,L6	2	2	3	2	3	2	2	1	2	1	–	2	2	2
CO5	L5	2	2	2	2	3	3	2	1	1	1	–	3	2	2

Textbooks

1. Singh R.C. and Rao N.R., Biology for Engineers, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.
2. Thyagarajan S. et al., Biology for Engineers, Tata McGraw-Hill, New Delhi, 2012.

Reference Books

1. Fox S. and Rompolski K., Human Physiology, McGraw-Hill eBook, 16th Edition, 2022.
2. Bar-Cohen Y., Biomimetics: Nature-Based Innovation, 1st Edition, CRC Press, 2012.
3. Ozbolat I., 3D Bioprinting: Fundamentals, Principles and Applications, Academic Press, 2016.

Web Resources

- NPTEL Biology for Engineers – https://nptel.ac.in/courses/121106008
- MIT OpenCourseWare – Introduction to Biological Engineering: https://ocw.mit.edu
- AlphaFold Structure Database – https://alphafold.ebi.ac.uk

SIGNALS, SYSTEMS AND STOCHASTIC PROCESSES

JNTUA R26 | II Year I Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04301T	II Year I Sem	2	0	2	0	3

Prerequisites: Engineering Mathematics, Network Analysis, Probability Theory

Course Objectives

- Understand the classification and mathematical representation of continuous-time and discrete-time signals and systems, and analyse them using Fourier series and Fourier transform techniques.
- Apply Laplace transform and its properties for system analysis, stability evaluation (BIBO), and transfer function determination.
- Understand the sampling theorem, aliasing, and the behaviour of LTI systems including distortionless transmission and Paley-Wiener criterion.
- Analyse random processes with respect to stationarity, ergodicity, autocorrelation, cross-correlation, Gaussian, and Poisson processes.
- Characterise random process spectral properties using power spectral density, cross-power density spectrum, and the Wiener-Khinchin theorem.

Course Content

UNIT I Signals & Systems

Core Content:

Basic definitions and classification of signals and systems (continuous-time and discrete-time), operations on signals, concepts of convolution and correlation of signals, analogy between vectors and signals — orthogonality, mean square error. Fourier series: trigonometric and exponential forms, properties, concept of discrete spectrum, illustrative problems.

AI Integration:

Use AI/LLM tools to explain signal classification and walk through a discrete convolution sum step by step; plot basic signals and applies time-shifting and scaling operations. compute the first three Fourier coefficients of a square wave using AI tools, sketch the partial reconstruction, and verify convergence.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Signals & Systems lectures.

UNIT II Fourier Transform & Laplace Transform

Core Content:

Fourier Transform: Definition, computation and properties for different types of signals and systems, inverse Fourier transform. Laplace Transform: Definition, region of convergence (ROC), properties, inverse Laplace transforms, the s-plane and BIBO stability, transfer functions, system response to standard signals, illustrative problems.

AI Integration:

Use AI/LLM tools to explain Fourier transform properties and the ROC concept, showing how pole locations determine BIBO stability; compute the Fourier transform of a rectangular pulse and shows how spectrum bandwidth changes with pulse width, verifying time-frequency duality.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL lectures.

UNIT III Sampling & Signal Transmission through Linear Systems

Core Content:

Sampling theorem — analytical proof for band-limited signals, reconstruction of signal from its samples, effect of under-sampling — aliasing, illustrative problems. Signal Transmission through Linear Systems: linear system, impulse response, response of a linear system for different input signals, LTI and LTV systems, transfer function of an LTI system; ideal LPF, HPF and BPF characteristics, distortionless transmission, bandwidth concepts, causality and Paley-Wiener criterion for physical realization.

AI Integration:

Use AI/LLM tools to explain the sampling theorem and aliasing with a 1 kHz sine wave example at different sampling rates; sketch spectra and identify aliased frequencies. sample a signal below and above the Nyquist rate for visual observation of aliasing; explain distortionless transmission and the Paley-Wiener criterion.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini.

UNIT IV Random Processes – Temporal Characteristics

Core Content:

The random process concept, classification of processes, deterministic and non-deterministic processes, distribution and density functions, stationarity and statistical independence. First-order stationary processes, second-order and wide-sense stationarity, time averages and ergodicity, autocorrelation function and its properties, cross-correlation function and its properties, covariance functions, Gaussian random processes, Poisson random process. Mean and mean-squared value of system response, autocorrelation function of response, cross-correlation functions of input and output.

AI Integration:

Use AI/LLM tools to explain stationarity and ergodicity using analogies and compare WSS and SSS processes with examples; simulate a Gaussian random process and compares sample statistics with theoretical values. cover key autocorrelation properties and work through a numerical autocorrelation example for a WSS process.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini.

UNIT V Random Processes – Spectral Characteristics

Core Content:

The power spectrum: properties, relationship between power spectrum and autocorrelation function, the cross-power density spectrum — properties, relationship between cross-power spectrum and cross-correlation function. Spectral characteristics of system response: power density spectrum of response, cross-power density spectra of input and output.

AI Integration:

Use AI/LLM tools to explain PSD and the Wiener-Khinchin theorem; sketch and compare PSD of white noise vs. a band-limited signal. compute PSD using FFT and verifies it against the Fourier transform of the autocorrelation function, confirming the Wiener-Khinchin relationship numerically.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL lectures.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Classify signals and systems, perform signal operations, and represent periodic signals using Fourier series with discrete spectrum interpretation.	L3 (Apply)
CO2	Compute Fourier and Laplace transforms, determine ROC and transfer functions, and evaluate BIBO stability of LTI systems.	L4 (Analyze)
CO3	Apply the sampling theorem to determine Nyquist rates, analyse aliasing effects, and characterise LTI system response and ideal filter properties.	L3 (Apply) L4 (Analyze)
CO4	Analyse random processes for stationarity, ergodicity, autocorrelation, and cross-correlation including Gaussian and Poisson process models.	L4 (Analyze)

CO	Course Outcome Statement	Bloom's Level
CO5	Compute power spectral density and cross-power density spectra using the Wiener-Khinchin theorem and determine spectral characteristics of system responses.	L4 (Analyze) L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	2	1	2	2	–	–	–	–	1	–	1	3	2
CO2	L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO3	L3,L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO4	L4	3	2	1	2	2	–	–	–	1	1	–	1	2	2
CO5	L4,L5	3	3	2	2	3	–	–	–	1	2	–	2	3	2

Textbooks

1. Peebles P.Z., Probability, Random Variables & Random Signal Principles, 4th Edition, TMH, 2002.
2. Oppenheim A.V., Willsky A.S., Nawab S.H., Signals and Systems, 2nd Edition, PHI, 2009.

Reference Books

1. Lathi B.P., Signals, Systems & Communications, BSP, 2013.
2. Papoulis A. and Pillai S.U., Probability, Random Variables and Stochastic Processes, 4th Edition, PHI, 2002.
3. Haykin S. and Van Veen, Signals & Systems, 2nd Edition, Wiley, 2005.
4. Hsu H., Schaum's Outline of Signals and Systems, 4th Edition, TMH, 2019.

ELECTRONIC DEVICES AND CIRCUITS

JNTUA R26 | II Year I Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04302T	II Year I Sem	2	0	2	0	3

Prerequisites: Physics for Communication Systems, Network Analysis, Engineering Mathematics

Course Objectives

- Understand the construction, operating principles and V-I characteristics of PN junction diode and special diodes.
- Analyse BJT characteristics in various configurations and understand biasing techniques.
- Analyse BJT small signal operation and models.
- Understand JFET and MOSFET device structures, V-I characteristics and biasing methods.
- Analyse MOSFET small-signal models and single-stage MOS amplifier configurations.

Course Content

UNIT I PN Junction Diode & Special Diodes

Core Content:

PN junction diode V-I characteristics, PN diode current equation, diode resistance, transition and diffusion capacitance, effect of temperature on PN junction diode. Half-wave, full-wave and bridge rectifiers with and without filters, ripple factor and regulation characteristics. Zener and avalanche breakdowns, V-I characteristics of Zener diode, Zener diode as voltage regulator, clipping and clamping circuits, illustrative problems. Special Diodes: Construction, operation and V-I characteristics of tunnel diode, LED, photo diode and UJT.

AI Integration:

Use AI/LLM tools (ChatGPT/Gemini) to explain the diode current equation, compare rectifiers (ripple factor, efficiency), and describe clipping/clamping circuits with waveform sketches. Plot the Shockley I-V curve in Python (Google Colab) varying temperature, and prepare a comparison table of special diodes (LED, photodiode, tunnel diode, UJT) covering structure, working principle, and applications.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini

UNIT II BJT Characteristics, Biasing & Stabilization

Core Content:

Review of bipolar junction transistors, characteristics, transistor as an amplifier and as a switch, BJT configurations, limits of operation, BJT specifications. Biasing and Stabilization: operating point, DC and AC load lines, importance of biasing, fixed bias, collector-to-base bias, self-bias, bias stability, thermal runaway, thermal stability, illustrative problems.

AI Integration:

Use AI/LLM tools to explain CE, CB, and CC configurations, find the Q-point for a fixed bias circuit, and compare fixed bias, collector-to-base bias, and self-bias in a table covering stability factor and Q-point stability. Plot the DC load line and Q-point in Python (Google Colab), varying the collector resistor to observe Q-point shift.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini

UNIT III BJT Small Signal Operation and Models

Core Content:

Transconductance, input resistance at the base and emitter, voltage gain, separating signal and DC quantities, the Hybrid-π model, the T model. Single stage BJT amplifiers: common-emitter (CE) amplifier without and with emitter resistance, common-base (CB) amplifier, common-collector (CC) amplifier, problem solving.

AI Integration:

Use AI/LLM tools to derive voltage gain for CE, CB, and CC amplifiers using the Hybrid-π and T-models, and compare the three configurations in a table showing gain, input resistance, and output resistance. Verify results using a Python (Google Colab) template for given gm and rπ values.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; ChatGPT / Gemini; NPTEL Microelectronics lectures

UNIT IV FET & MOSFET

Core Content:

Junction Field Effect Transistor (JFET): construction, principle of operation, V-I characteristics, comparison of BJT and FET, FET as voltage variable resistor. MOS Field Effect Transistors: introduction, device structure and physical operation, CMOS, V-I characteristics, MOSFET circuits at DC, MOSFET as an amplifier and as a switch. Biasing in MOS amplifier circuits — biasing by fixing VGS with and without source resistance, biasing using drain-to-gate feedback resistor, problem solving.

AI Integration:

Use AI/LLM tools to compare JFET and MOSFET (structure, control mechanism, biasing) and explain MOSFET biasing methods step by step. Plot MOSFET ID-VDS curves for three VGS values using the square-law model in Python (Google Colab), marking triode, saturation, and cutoff regions.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini

UNIT V MOSFET Small Signal Operation

Core Content:

DC bias, separating DC analysis and signal analysis, small signal equivalent circuit models, transconductance, the T equivalent circuit model. Single stage MOS amplifiers: common source (CS) amplifier without and with source resistance, common gate (CG) amplifier, source follower, problem solving.

AI Integration:

Use AI/LLM tools to derive voltage gain for CS, CG, and source follower configurations and compare them in a table showing gain, input/output resistance, and best-fit applications. Compute parameters in Python (Google Colab) for given gm and rd values, and explain the role of coupling and bypass capacitors in separating DC and AC analysis.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; ChatGPT / Gemini; NPTEL Microelectronics lectures

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Analyse PN junction diode characteristics, design rectifiers, clipping and clamping circuits, and describe the operation of special diodes.	L4 (Analyze)
CO2	Apply BJT biasing techniques to establish a stable Q-point, analyse DC/AC load lines, and evaluate thermal stability.	L3 (Apply) L4 (Analyze)
CO3	Apply BJT Hybrid-π and T-models to compute voltage gain and input/output impedance of single-stage amplifiers.	L3 (Apply) L4 (Analyze)
CO4	Analyse JFET and MOSFET V-I characteristics, evaluate different biasing methods, and design MOSFET circuits for amplification and switching.	L4 (Analyze)
CO5	Apply MOSFET small-signal models to determine gain and impedance of CS, CG, and source follower configurations and compare their performance.	L3 (Apply) L4 (Analyze)

CO – PO Articulation Matrix

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

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

Textbooks

1. Sedra A.S. and Smith K.C., Microelectronic Circuits, 6th Edition, Oxford Press, 2013.
2. Millman J. and Halkias C., Integrated Electronics, 2nd Edition, Tata McGraw Hill, 1991.

Reference Books

1. Neamen D.A., Electronic Circuits – Analysis and Design, 3rd Edition, McGraw Hill, 2019.
2. Razavi B., Microelectronics, 2nd Edition, Wiley, 2013.
3. Boylestad R.L. and Nashelsky L., Electronic Devices and Circuits, 9th Edition, Pearson, 2006.
4. Streetman B.G. and Banerjee S., Solid State Electronic Devices, 7th Edition, Pearson.

DIGITAL CIRCUITS DESIGN

JNTUA R26 | II Year I Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04303T	II Year I Sem	2	0	2	0	3

Prerequisites: Digital Logic and Computer Organization, Engineering Mathematics

Course Objectives

- Understand number systems, Boolean algebra laws, and minimisation of Boolean functions using Karnaugh maps for logic circuit design.
- Design and verify combinational logic circuits including arithmetic circuits, data routing circuits, code converters, and parity circuits.
- Understand and apply Verilog HDL for structural, dataflow, and behavioural description of combinational and sequential digital circuits.
- Analyse and design sequential logic circuits including flip-flops, counters, and shift registers using state diagrams and excitation tables.
- Design Finite State Machines for sequence detection and understand the structure and advantages of PLDs and FPGAs.

Course Content

UNIT I Number Systems, Codes & Boolean Algebra

Core Content:

Binary, octal, decimal, and hexadecimal number systems, number conversions, binary arithmetic, signed binary numbers, 1's and 2's complements. BCD, Gray, Excess-3, and ASCII codes. Boolean algebra postulates and theorems, De Morgan's theorems, Boolean functions, SOP and POS forms, canonical forms, and algebraic minimization.

AI Integration:

Students use AI/LLM tools to solve and verify number conversions and signed binary arithmetic problems. A Python (Google Colab) template is used to convert decimal to binary/octal/hex and compare canonical SOP, POS, and simplified Boolean expressions.

Free/Open-Source AI Tools:

Python (NumPy, SymPy) on Google Colab; Logisim; online Boolean calculators; ChatGPT / Gemini

UNIT II Logic Gates & Gate Level Minimisation

Core Content:

AND, OR, NOT, NAND, NOR, XOR, and XNOR gates — truth tables, logic symbols, and Boolean expressions; gate-level realisation of Boolean expressions. Karnaugh map minimization for 2-variable, 3-variable, 4-variable, and 5-variable functions; SOP and POS minimization with don't-care conditions; prime implicants and essential prime implicants; NAND-only and NOR-only implementations of minimized expressions.

AI Integration:

Students use AI/LLM tools to understand prime implicants and NAND/NOR-only implementations through worked examples. A Python K-map template auto-groups minterms and verifies minimized SOP expressions against manual solutions.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; Logisim; K-map solver tools online; ChatGPT / Gemini

UNIT III Combinational Circuits

Core Content:

Half adder, full adder, half subtractor, full subtractor, parallel binary adder, BCD adder, magnitude comparator, BCD to Excess-3 converter, binary to Gray converter, multiplexers, demultiplexers, encoders, and decoders. Introduction to HDL, Verilog/VHDL basics, dataflow and behavioural modelling; HDL programs for logic gates, adders, multiplexers, decoders, and code converters.

AI Integration:

Students use AI/LLM tools to derive adder and BCD adder logic and to write and trace a Verilog program for a BCD-to-Excess-3 converter. A 4-to-1 MUX and 2-to-4 decoder are simulated in Python (Google Colab) to verify Boolean function implementation.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; Logisim; EDA Playground (free Verilog simulation); ChatGPT / Gemini

UNIT IV Sequential Circuits

Core Content:

SR latch, D latch, SR, JK, D, and T flip-flops, master-slave flip-flops, and flip-flop conversions. SISO, SIPO, PISO, PIPO, and universal shift registers. Asynchronous counters, synchronous counters, up/down counters, modulo-N counters, ring counters, and Johnson counters. HDL modelling of flip-flops, registers, counters, and test bench development.

AI Integration:

Students use AI/LLM tools to work through flip-flop excitation tables and JK-to-D conversion, then write a Verilog behavioural model with test bench for a JK flip-flop and ring counter. A 4-bit synchronous up-counter and SIPO shift register are simulated in Python to verify state sequences.

Free/Open-Source AI Tools:

Python (NumPy) on Google Colab; Logisim; EDA Playground (free Verilog simulation); ChatGPT / Gemini

UNIT V Sequential Design & Programmable Logic Devices

Core Content:

State diagrams, state tables, state assignment, design of synchronous sequential circuits, Mealy and Moore sequence detectors. HDL implementation of sequence detectors, finite state machines, and simple ALU design. PLA, PAL, PROM, and FPGA — structure, programming tables, and implementation.

AI Integration:

Students use AI/LLM tools to design a '1011' sequence detector as both Mealy and Moore machines and compare PLA, PAL, PROM, and FPGA architectures. A Verilog FSM for a sequence detector and a basic ALU are written, with exposure to the FPGA design flow from HDL to bitstream.

Free/Open-Source AI Tools:

Python on Google Colab; EDA Playground; Intel Quartus Prime Lite / Xilinx Vivado (free); ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply number systems, binary codes, Boolean algebra, and De Morgan's theorems to represent, convert, and simplify digital logic functions in SOP and POS forms.	L3 (Apply)
CO2	Minimise Boolean functions using K-maps with don't-care conditions; identify prime implicants and implement using NAND-only and NOR-only gate networks.	L3 (Apply) L4 (Analyze)
CO3	Design combinational circuits — arithmetic units, code converters, MUX/DEMUX, encoders, and decoders; model and simulate using Verilog/VHDL.	L3 (Apply) L4 (Analyze)
CO4	Design sequential circuits — flip-flops, shift registers, and counters; develop HDL behavioural models and test benches for simulation.	L4 (Analyze)
CO5	Design synchronous sequential circuits using Mealy and Moore models; implement on PLA, PAL, CPLD, and FPGA platforms using HDL.	L5 (Evaluate) L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	1	2	–	–	–	1	1	–	1	3	2
CO2	L3,L4	3	3	2	2	2	–	–	–	1	1	–	1	3	2
CO3	L3,L4	3	3	3	2	3	–	–	–	1	2	–	2	3	2

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO4	L4	3	3	2	2	3	–	–	–	1	1	–	2	3	2
CO5	L5,L6	3	3	3	2	3	–	–	–	1	2	1	2	3	3

Textbooks

1. Mano M.M. and Ciletti M.D., Digital Design, 5th Edition, Pearson Education, 2013.
2. Jain R.P., Modern Digital Electronics, 4th Edition, Tata McGraw Hill, 2009.

Reference Books

1. Givone D.D., Digital Principles and Design, Tata McGraw Hill, 2002.
2. Roth C.H. and Kinney L.L., Fundamentals of Logic Design, 7th Edition, Cengage Learning, 2014.
3. Rao C.V.S., Switching and Logic Design, 3rd Edition, Pearson Education, 2009.
4. Kohavi Z. and Jha N.K., Switching and Finite Automata Theory, 3rd Edition, Cambridge University Press, 2010.

SIGNALS & SYSTEMS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04301P	II Year I Sem	0	0	0	2	1	Engineering Course Lab

Prerequisites: Engineering Mathematics; Basic Electronics; concurrent: Signals & Systems theory

Course Objectives

- Introduce signal and system concepts including transforms, LTI properties, stability, and filter response techniques using MATLAB/Simulink or Python.
- Develop understanding of frequency-domain analysis through Fourier series, Fourier transforms, and spectral interpretation of continuous and discrete-time signals.
- Familiarise students with sampling theory, noise analysis, and correlation-based techniques for signal processing and system characterisation.
- Enable students to design, simulate, and verify system properties such as linearity, time-invariance, causality, and stability using MATLAB/Simulink.
- Familiarise students with ML-based approaches and LLM tools for signal classification, system identification, parameter estimation, and automated analysis.

List of Experiments

Implement / execute any 12 experiments. All experiments are to be simulated using MATLAB or equivalent software.

Part A – Regular Experiments

1. Write a program to generate various Signals and Sequences: Periodic and Aperiodic, Unit Impulse, Unit Step, Square, Sawtooth, Triangular, Sinusoidal, Ramp, and Sinc function.
2. Perform operations on Signals and Sequences: Addition, Multiplication, Scaling, Shifting, Folding, Computation of Energy and Average Power.
3. Write a program to find the trigonometric and exponential Fourier series coefficients of a rectangular periodic signal. Reconstruct the signal by combining the Fourier series coefficients with appropriate weightings and plot the discrete spectrum of the signal.
4. Write a program to find the Fourier Transform of a given signal. Plot its amplitude and phase spectrum.
5. Write a program to convolve two discrete-time sequences. Plot all the sequences.
6. Write a program to find autocorrelation and cross-correlation of given sequences.
7. Write a program in MATLAB and/or Simulink to verify Linearity and Time Invariance properties of a given Continuous System.

8. Write a program in MATLAB and/or Simulink to generate a discrete-time sequence by sampling a continuous-time signal. Show that with sampling rates less than the Nyquist rate, aliasing occurs while reconstructing the signal.
9. Write a program in MATLAB and/or Simulink to find the magnitude and phase response of a first-order Low Pass and High Pass filter. Plot the responses in logarithmic scale.
10. Write a program to generate Complex Gaussian noise and find its mean, variance, Probability Density Function (PDF) and Power Spectral Density (PSD).
11. Generate Random data (with bipolar) for a given data rate (say 10 kbps). Plot the same for a time period of 0.2 sec.
12. Write a program to plot the pole-zero diagram in the S-plane for a given signal/sequence and verify its stability.
13. Write a program in MATLAB and/or Simulink to find the impulse response and step response of a system from its difference equation.
14. Write a program to remove noise by autocorrelation/cross-correlation in a given signal corrupted by noise.
15. Build a system that amplifies a sine wave by a factor of two using Simulink.
16. Test the linearity of the system using Simulink.

Part B – AI / LLM and Python / MATLAB Tasks

Any six experiments need to be implemented using appropriate AI/LLM tools and/or Python/MATLAB-based simulations.

Experiment 1 – Signal Generation & Classification

a) Use AI/LLM tools to classify the generated signals as periodic/aperiodic and even/odd, and explain signal properties with examples.
b) Train a KNN classifier (Scikit-learn) using extracted features (mean, variance) to automatically classify the signal type.

Experiment 2 – Signal Operations

c) Use AI/LLM tools to predict the output of each operation before simulation and provide a conceptual explanation of the transformations.
d) Use Linear Regression to approximate the transformed signal values and compare with the simulated output.

Experiment 3 – Fourier Series

e) Use AI/LLM tools to give a step-by-step explanation of the Fourier series derivation and interpret the discrete spectrum.
f) Train a regression model to predict peak amplitude or output signal length as a function of the number of harmonics included.

Experiment 4 – Fourier Transform

g) Use AI/LLM tools to interpret spectral similarity or dissimilarity between two signals based on their amplitude and phase spectra.
h) Train a binary classifier to detect whether two signals are similar or dissimilar based on their spectral features.

Experiment 5 – Convolution

i) Use AI/LLM tools to explain the harmonic contribution of each input sequence to the convolution output.
j) Build a regression model to predict reconstruction error as a function of the number of harmonics used.

Experiment 6 – Correlation

k) Use AI/LLM tools to interpret frequency components and explain the significance of peaks in the correlation output.
l) Train a classification model to identify the signal type from its correlation-based spectrum features.

Experiment 7 – LTI Properties

m) Use AI/LLM tools to verify and explain whether the system satisfies superposition and homogeneity, and reason about LTI properties.
n) Train a Decision Tree classifier to classify the system as LTI or Non-LTI based on input-output data pairs.

Experiment 8 – Sampling & Aliasing

o) Use AI/LLM tools to predict aliasing conditions and explain the implications of the Nyquist sampling theorem.
p) Build a binary classifier (Logistic Regression) to detect aliasing from sampling rate and signal frequency parameters.

Experiment 9 – Filter Response

q) Use AI/LLM tools to predict and explain filter response behaviour, including cutoff frequency and bandwidth characteristics.
r) Train a regression model to estimate the cutoff frequency from filter design parameters.

Experiment 10 – Gaussian Noise

s) Use AI/LLM tools to explain the statistical properties of Gaussian noise, the significance of PSD, and the interpretation of the PDF.
t) Apply a Gaussian Naive Bayes classifier to identify the distribution type from computed statistical features.

Experiment 11 – Random Bipolar Data

u) Use AI/LLM tools to explain the statistical and spectral properties of bipolar random data and their significance in digital communications.
v) Apply a Gaussian Naive Bayes classifier to identify the distribution type of the generated random data.

Experiment 12 – Pole-Zero & Stability

w) Use AI/LLM tools to suggest an appropriate filtering technique based on the pole-zero locations and stability analysis.
x) Build a simple denoising model using Linear Regression or ML-based Moving Average tuning to reduce noise.

Experiment 13 – Impulse & Step Response

y) Use AI/LLM tools to explain the relationship between the difference equation and the system response (impulse & step), including stability and causality interpretation.

z) Train a regression model (Linear Regression / MLP Regressor) to predict the impulse or step response values based on input coefficients.

Experiment 14 – Noise Removal

aa) Use AI/LLM tools to explain how autocorrelation and cross-correlation help in identifying useful signal components and reducing noise.

bb) Train a regression model (Linear Regression / Ridge Regression) to estimate the clean signal from a noisy input, or use a basic denoising autoencoder.

Experiment 15 – Sine Wave Amplification (Simulink)

cc) Use AI/LLM tools to predict and explain the effect of amplification (gain = 2) on amplitude, power, and frequency characteristics of the sine wave.

dd) Train a regression model to learn the relationship between input amplitude and output amplitude (gain modeling).

Experiment 16 – Linearity Test (Simulink)

ee) Use AI/LLM tools to verify and explain whether the system satisfies superposition and homogeneity conditions for various test inputs.

ff) Train a binary classification model (Logistic Regression / Decision Tree) to classify the system as Linear or Non-Linear based on input-output data.

Note: The suggested AI/LLM and/or Python/MATLAB tasks can be modified based on the decision of the subject expert.

AI Integration	AI/ML integration is embedded across all 16 experiments in two tiers: (i) AI/LLM tools (ChatGPT, Claude.ai, Gemini) are used to provide conceptual explanations, predict outcomes, and interpret results – building analytical reasoning alongside simulation; (ii) Python/Scikit-learn ML models (KNN, Decision Tree, Logistic Regression, Linear/Ridge Regression, MLP, Naive Bayes, denoising autoencoder) are trained on simulation outputs to automate signal classification, system identification, aliasing detection, filter parameter estimation, noise reduction, and gain modeling. Any six of the sixteen AI/ML task pairs must be completed, with choice left to the subject expert.

Tools & Platforms: MATLAB / Simulink; Python (NumPy, SciPy, Matplotlib, Scikit-learn) – Google Colab (free); ChatGPT / Claude.ai / Gemini – LLM assistance; LibreOffice Calc – data recording.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Simulate and analyse continuous and discrete-time signals using MATLAB/Python; apply signal operations, transforms, and spectral analysis to characterise signal properties.	L3 (Apply)

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO2	Verify LTI system properties, stability, and causality through impulse/step response analysis and pole-zero diagrams using MATLAB/Simulink.	L4 (Analyse)
CO3	Evaluate system performance using frequency-domain analysis, sampling theory, noise characterisation, and correlation-based noise reduction techniques.	L4–L5 (Analyse, Evaluate)
CO4	Design and simulate filter responses, random signal generation, and amplification systems using MATLAB/Simulink, validating results against theoretical expectations.	L5 (Evaluate)
CO5	Build and apply ML classifiers and regression models using Python/Scikit-learn to automate signal classification, system identification, aliasing detection, and parameter estimation, leveraging LLM tools for result interpretation.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO5	Define LTI system; state Nyquist theorem; recall convolution formula; define autocorrelation; state Fourier transform properties
L2 – Understand	Explain concepts, describe procedures	CO1–CO5	Explain aliasing; describe pole-zero stability; explain Fourier transform phase spectrum; describe KNN classification; explain filter cutoff behaviour
L3 – Apply	Simulate, implement, compute	CO1	Simulate signal generation in MATLAB; compute energy and power; implement correlation; plot magnitude/phase response; generate Gaussian noise and compute PDF/PSD
L4 – Analyse	Compare, verify, interpret	CO2, CO3	Verify LTI properties via superposition/homogeneity; analyse pole-zero locations for stability; compare autocorrelation output with noise-free signal; interpret spectral similarity
L5 – Evaluate	Assess, validate models	CO3, CO4, CO5	Evaluate aliasing conditions vs. Nyquist theorem; validate filter response against design specs; assess ML model R^2/RMSE; evaluate classifier accuracy; compare LLM prediction with simulation
L6 –	Build,	CO5	Build KNN/Decision Tree classifiers for signal type;

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
Create	automate, develop		develop gain regression model; create denoising autoencoder; implement Logistic Regression for linearity/aliasing detection; design Simulink amplification system

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
CO1	L3	3	2	2	1	2	–	1	1	2	2	–	2	3
CO2	L4	3	3	2	1	2	–	1	1	2	2	–	2	3
CO3	L4-L5	3	3	2	1	3	1	1	1	2	2	–	2	3
CO4	L5	3	2	3	1	3	1	1	1	2	2	–	2	3
CO5	L5-L6	2	3	3	1	3	1	1	1	2	2	1	3	2

Recommended Tools & Platforms

Recommended Free & Open-Source AI / Computational Tools
MATLAB / Simulink – Signal generation, LTI verification, filter response, sampling, Simulink-based amplification and linearity testing
Python (NumPy, SciPy, Matplotlib, Scikit-learn) on Google Colab (free) – signal processing, ML classification and regression models
ChatGPT / Claude.ai / Gemini – LLM-based conceptual explanation, result interpretation, and prediction of system behaviour
Scikit-learn (free, open-source) – KNN, Decision Tree, Logistic Regression, Linear/Ridge Regression, MLP Regressor, Gaussian Naive Bayes
LibreOffice Calc (free, open-source) – standardised data recording templates for all experiments

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed

Assessment Component	**Frequency / Duration**	**Marks**	**COs Assessed**
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI / ML Lab Exercises (any 6)	6 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs (CO1–CO5)
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI/ML exercises to be submitted as Google Colab .ipynb notebooks.

II Year I Sem

ELECTRONIC DEVICES AND CIRCUITS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04302P	II Year I Sem	0	0	0	3	1.5	Programme Core Lab – ECE

Prerequisites: Engineering Physics; Basic Electronics; concurrent: Electronic Devices and Circuits theory

Course Objectives

- Introduce V-I and transfer characteristics of diodes (PN junction, Zener), UJT, BJT, JFET, and MOSFET, and the methods to extract device parameters from experimental data.
- Develop understanding of rectifier circuits (half wave and full wave), clipping and clamping circuits, and their performance metrics such as ripple factor and efficiency.
- Enable students to design and analyse biasing networks (voltage-divider bias, self-bias) for BJT and MOSFET, and determine the stable Q-point for linear amplifier operation.
- Familiarise students with the design and frequency response analysis of small signal amplifiers using BJT (CE) and MOSFET (CS), including RC coupled amplifier configurations.
- Familiarise students with LLM-based circuit interpretation and ML/Python techniques for device parameter estimation, operating point prediction, and performance curve analysis.

List of Experiments

Implement / execute any 12 experiments.

Part A – Regular Experiments

- Plot V-I characteristics of a PN Junction diode.
- Plot V-I characteristics of a Zener Diode and demonstrate its application as a Voltage Regulator.
- Implement half wave and full wave rectifiers, observe the output waveforms, and analyse the ripple factor and efficiency.
- Verify various clipping and clamper circuits using a PN junction diode and draw suitable graphs for each configuration.
- Draw the V-I characteristics of a Unijunction Transistor (UJT) and determine the parameters η, Ip, Iv, Vp, and Vv.
- Verify the input and output characteristics of a BJT in Common Emitter (CE) configuration and determine required parameters from the graphs.

- Verify the input and output characteristics of a BJT in Common Base (CB) configuration and determine required parameters from the graphs.
- Draw the V-I characteristics of a Junction Field Effect Transistor (JFET) and extract key parameters.
- Draw the output and transfer characteristics of a MOSFET (Enhancement mode and Depletion mode) in CS configuration and determine required parameters from the graphs.
- Design and analyse a voltage-divider bias / self-bias circuit using BJT and determine the Q-point.
- Design and analyse a self-bias circuit using MOSFET and determine the operating point.
- Design a small signal amplifier using MOSFET (common source) for given specifications. Draw the frequency response and find the bandwidth.
- Design a small signal amplifier using BJT (common emitter) for given specifications. Draw the frequency response and find the bandwidth.
- Design and analyse an RC coupled amplifier for given design parameters and determine its frequency response.

Part B – AI / LLM and Python / MATLAB Tasks

Any six experiments need to be implemented using appropriate AI/LLM tools and/or Python/MATLAB-based simulations.

Experiment 1 – PN Junction Diode V-I Characteristics

- Use AI/LLM tools to interpret the forward bias, reverse bias, and breakdown regions of the V-I curve and explain the significance of threshold voltage and reverse saturation current.
- Train a regression model to fit the diode equation (Shockley equation) to the measured V-I data and estimate saturation current and ideality factor.

Experiment 2 – Zener Diode Characteristics & Voltage Regulator

- Use AI/LLM tools to explain Zener breakdown mechanism, the difference between avalanche and Zener breakdown, and how the diode maintains voltage regulation.
- Build a regression model to predict regulation performance (output voltage vs. load current) from measured characteristic data.

Experiment 3 – Half Wave & Full Wave Rectifiers

- Use AI/LLM tools to explain the working principle of half wave and full wave rectifiers, the role of the filter capacitor, and derive expressions for ripple factor and efficiency.
- Use Python/MATLAB simulation to predict ripple voltage and efficiency; apply regression to model ripple factor as a function of load and capacitance.

Experiment 4 – Clipping & Clamping Circuits

- Use AI/LLM tools to predict the output waveform shape for each clipping and clamping configuration before hardware verification, and explain the role of diode polarity and reference voltage.
- Train a binary classifier to identify the circuit type (clipper/clamper, positive/negative) from the output waveform features.

Experiment 5 – UJT Characteristics

- Use AI/LLM tools to explain the UJT structure, intrinsic standoff ratio, and the significance of the peak and valley points in the V-I characteristics.
- Apply curve fitting (polynomial regression) to the V-I data to accurately estimate the parameters η, Ip, Iv, Vp, and Vv.

Experiment 6 – BJT CE Characteristics

- Use AI/LLM tools to explain the physical basis of CE input/output characteristics, the concept of current gain β, and the Early effect.
- Train a regression model on the output characteristic data to estimate the current gain β and output resistance from the measured curves.

Experiment 7 – BJT CB Characteristics

- Use AI/LLM tools to explain CB configuration characteristics, compare them with CE configuration, and reason about the low input impedance and near-unity current gain of CB.
- Build a regression or parameter prediction model to estimate α, input resistance, and output resistance from CB characteristic data.

Experiment 8 – JFET Characteristics

- Use AI/LLM tools to explain the I-V behaviour of JFET in the ohmic, saturation, and pinch-off regions, and the significance of IDSS and Vp.
- Apply regression to model the drain current as a function of VGS and VDS, and estimate key JFET parameters.

Experiment 9 – MOSFET Characteristics (Enhancement & Depletion)

- Use AI/LLM tools to explain the differences between enhancement and depletion mode MOSFETs, the concept of threshold voltage, and the body effect.
- Train a regression/classification model to predict the threshold voltage from transfer characteristic data points.

Experiment 10 – BJT Voltage-Divider Bias / Q-Point

- Use AI/LLM tools to explain the voltage divider bias technique, DC load line concept, and how the Q-point ensures linear amplifier operation.
- Build a simple ML-based Q-point predictor that estimates ICQ and VCEQ from circuit component values.

Experiment 11 – MOSFET Self-Bias Operating Point

- Use AI/LLM tools to explain MOSFET self-bias operation, the role of the source resistor in stabilising the operating point, and compare with BJT biasing.
- Train a regression model to predict the MOSFET operating point (IDQ, VDSQ) from component values and MOSFET parameters.

Experiment 12 – MOSFET Common Source Amplifier

- Use AI/LLM tools to explain gain-bandwidth trade-off in MOSFET amplifiers, the effect of coupling and bypass capacitors on frequency response, and the concept of −3 dB bandwidth.
- Apply regression to model and predict the gain-bandwidth product from design parameters such as gm, RD, and capacitor values.

Experiment 13 – BJT Common Emitter Amplifier

- Use AI/LLM tools to analyse the BJT CE amplifier frequency response, explain low-frequency and high-frequency cutoff causes, and interpret the Bode plot.
- Use curve fitting (polynomial or spline regression) to model the frequency response and estimate the 3 dB bandwidth automatically from simulation data.

Experiment 14 – RC Coupled Amplifier

- Use AI/LLM tools to explain RC coupling, the purpose of coupling and bypass capacitors, inter-stage loading effects, and how they shape the frequency response.
- Combine LLM-guided design with a regression/optimisation model to predict midband gain and bandwidth for different RC values, enabling design optimisation.

Note: The suggested AI/LLM and/or Python/MATLAB tasks can be modified based on the decision of the subject expert.

AI Integration: AI/ML integration is embedded across all 14 experiments in two tiers: (i) AI/LLM tools (ChatGPT, Claude.ai, Gemini) provide conceptual interpretation of device characteristics, circuit behaviour prediction, and Bode plot analysis – bridging theory with observation; (ii) Python/Scikit-learn ML models (Shockley equation regression, polynomial curve fitting, binary classifiers, Q-point predictors, gain-bandwidth regression, spline/polynomial frequency response fitting) are applied to measured or simulated data to automate parameter extraction, operating point estimation, and performance prediction. Any six of the fourteen AI/ML task pairs must be completed, at the subject expert's discretion.

Software Tools: Multisim / PSpice or equivalent; Python (NumPy, SciPy, Matplotlib, Scikit-learn) – Google Colab (free); ChatGPT / Claude.ai / Gemini – LLM assistance.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Plot and interpret V-I and transfer characteristics of PN junction diode, Zener diode, UJT, BJT, JFET, and MOSFET; extract device parameters such as β, α, IDSS, Vp, Vth, η, Ip, and Iv from experimental curves.	L3 (Apply)
CO2	Analyse rectifier circuits and waveform shaping circuits (clippers and clampers), measuring ripple factor, efficiency, and output waveform accuracy for different configurations.	L4 (Analyse)
CO3	Design and evaluate BJT and MOSFET biasing networks (voltage-divider bias and self-bias), determine the DC load line and Q-point, and assess the stability of the operating point.	L4–L5 (Analyse, Evaluate)
CO4	Design small signal amplifiers using BJT (CE) and MOSFET (CS), plot and interpret the frequency response (Bode plot), and determine bandwidth, midband gain, and gain-bandwidth product.	L5 (Evaluate)

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO5	Apply ML regression models and classifiers using Python/Scikit-learn to fit device equations, predict operating points, estimate circuit parameters, and automate characteristic curve analysis, leveraging LLM tools for conceptual interpretation.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO5	Define threshold voltage; state Shockley equation; recall rectifier efficiency formula; define Q-point; state Nyquist criterion for amplifier stability
L2 – Understand	Explain concepts, describe procedures	CO1–CO5	Explain Zener breakdown vs. avalanche; describe MOSFET body effect; explain voltage divider bias stability; describe gain-bandwidth trade-off; explain RC coupling role
L3 – Apply	Plot, measure, compute	CO1	Plot V-I characteristics of PN diode, Zener, UJT, BJT (CE/CB), JFET, MOSFET; extract β, α, IDSS, Vp, η from graphs; measure ripple factor and efficiency
L4 – Analyse	Compare, verify, interpret	CO2, CO3	Analyse clipper/clamper output waveforms; compare CE vs. CB parameters; verify Q-point stability under load variation; analyse DC load line for biasing circuits
L5 – Evaluate	Assess, validate, optimise	CO3, CO4, CO5	Evaluate Q-point stability of BJT vs. MOSFET bias; validate amplifier bandwidth against design specs; assess regression model accuracy (R^2/RMSE); evaluate gain-bandwidth product optimisation
L6 – Create	Design, build, automate	CO5	Design CE/CS amplifier for given specs; build Shockley regression model; develop Q-point ML predictor; create waveform classifier (clipper/clamper); automate 3 dB bandwidth extraction via spline fitting

CO – PO / PSO Articulation Matrix (NBA)

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
CO1	L3	3	2	2	1	2	–	1	1	2	1	–	1	3
CO2	L4	3	3	2	1	2	1	1	1	2	1	–	2	3
CO3	L4-L5	3	3	3	1	2	1	1	1	2	2	–	2	3

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
CO4	L5	3	2	3	1	3	1	1	1	2	2	1	2	3
CO5	L5-L6	2	3	3	1	3	1	1	1	2	2	1	3	2

Recommended Tools & Platforms

- Multisim / PSpice / LTspice – circuit simulation for V-I characteristics, rectifiers, clippers, biasing circuits, and amplifier frequency response
- Python (NumPy, SciPy, Matplotlib, Scikit-learn) on Google Colab (free) – Shockley regression, Q-point ML predictor, waveform classifier, bandwidth estimation
- ChatGPT / Claude.ai / Gemini – LLM-based conceptual interpretation of device characteristics, circuit behaviour prediction, and Bode plot analysis
- Scikit-learn (free, open-source) – polynomial regression, binary classifiers, Linear/Ridge Regression, MLP Regressor for device parameter estimation
- LibreOffice Calc (free, open-source) – standardised data recording templates, graph plotting for all experiments

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI / ML Lab Exercises (any 6)	6 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs (CO1–CO5)
TOTAL		100	

Note: Lab record to be submitted for every experiment at the next session. AI/ML exercises to be submitted as Google Colab .ipynb notebooks.

DIGITAL CIRCUITS DESIGN LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04303P	II Year I Sem	0	0	0	3	1.5	Programme Core Lab – ECE

Prerequisites: Digital Electronics theory; Basic Logic Design; concurrent: Digital Circuits Design theory

Course Objectives

- Introduce the design and verification of basic and universal logic gates, and combinational digital circuits including adders, comparators, code converters, decoders, and multiplexers.
- Develop understanding of Boolean minimisation techniques (K-map, SOP/POS) and their application in realising efficient combinational logic circuits.
- Enable students to design and verify sequential circuits including flip flops (SR, JK, D), counters (ripple and synchronous), shift registers, and ring counters using truth tables, state diagrams, and excitation tables.
- Familiarise students with Verilog HDL for digital system design and simulation using EDA tools such as ModelSim or Vivado.
- Familiarise students with LLM tools for truth table generation, logic derivation, and Verilog debugging, and Python/ML techniques for state prediction, waveform verification, and Boolean minimisation automation.

List of Experiments

Implement / execute any 12 experiments.

Part A – Regular Experiments

1. Verify the operation of Basic Gates (AND, OR, NOT) and Universal Gates (NAND, NOR) using their respective Truth Tables.
2. Design a simple combinational circuit with four variables, obtain the minimal SOP expression using Karnaugh Map, and verify the truth table using a Digital Trainer Kit.
3. Verify the functional table of a 3-to-8 line Decoder / Demultiplexer IC.
4. Implement a four-variable logic function using an 8-to-1 Multiplexer and verify the output.
5. Design a full adder circuit using logic gates and verify its functional table.
6. Verify the Truth Table of S-R, J-K, and D flip flops using respective ICs and observe timing diagrams.
7. Design a 4-bit ring counter using D Flip-Flops or JK Flip-Flops and verify the output sequence.
8. Verify the operation of a 4-bit Universal Shift Register for different modes of operation (SISO, SIPO, PISO, PIPO).

9. Draw the circuit diagram of a MOD-8 ripple counter, construct the circuit using T Flip-Flops, test with a low-frequency clock, and sketch the output waveforms.
10. Design a MOD-8 synchronous counter using T Flip-Flops, verify the result, and sketch the output waveforms.
11. Draw the circuit diagram of a single-bit comparator, construct the circuit, and test the output for all input combinations.
12. Construct a 7-Segment Display circuit using a BCD-to-7-segment Decoder and 7-Segment LED, and test it for all BCD inputs.
13. Design and realise a BCD to Excess-3 code converter and vice versa using combinational logic.
14. Design a simple decision-making digital system using Verilog HDL. Simulate and verify using ModelSim or Vivado.

Part B – AI / LLM and Python / MATLAB Tasks

Any six experiments need to be implemented using appropriate AI/LLM tools and/or Python/MATLAB-based simulations.

Experiment 1 – Basic & Universal Gates

a) Use AI/LLM tools to generate the truth tables for all basic and universal gates, verify logical equivalence (e.g., NAND as universal gate), and explain De Morgan’s theorems.

b) Use AI tools or Python (SymPy) to simplify Boolean expressions and verify logical equivalence between gate combinations programmatically.

Experiment 2 – K-Map Minimisation

c) Use AI/LLM tools to assist in K-map grouping, derive the minimal SOP/POS expressions, and verify the simplification step by step.

d) Use Python (SymPy or custom logic) to automate SOP/POS minimisation and validate the reduced expression against the original truth table.

Experiment 3 – 3-to-8 Decoder / Demultiplexer

e) Use AI/LLM tools to explain the working of a 3-to-8 decoder, describe its enable logic, and generate the complete functional table automatically.

f) Use Python to automate truth table generation for any N-to-2! decoder and verify against hardware observations.

Experiment 4 – 8-to-1 Multiplexer

g) Use AI/LLM tools to explain how a multiplexer implements logic functions, derive the input mapping for the given four-variable function, and verify the MUX configuration.

h) Build a Python-based AI-assisted tool to automatically map any given logic function to MUX inputs using truth table analysis.

Experiment 5 – Full Adder

i) Use AI/LLM tools to derive the Boolean equations for sum and carry of a full adder, explain the role of the half adder stages, and verify with truth table walkthrough.

j) Simulate the full adder in Python/MATLAB and verify output against the expected truth table; use a classifier to validate correct gate combinations.

Experiment 6 – SR, JK & D Flip Flops

k) Use AI/LLM tools to explain the operation of S-R, J-K, and D flip flops, interpret timing diagrams, and reason about the forbidden state in S-R and race-around condition in J-K flip flops.

l) Train a simple binary classifier to predict the next state of a flip flop from its current state and input, based on truth table data.

Experiment 7 – 4-Bit Ring Counter

m) Use AI/LLM tools to explain the ring counter state sequence, distinguish it from a Johnson counter, and reason about self-correction and lockout states.

n) Use Python to model the ring counter state machine and implement a simple pattern classifier to predict the next state in the sequence.

Experiment 8 – 4-Bit Universal Shift Register

o) Use AI/LLM tools to explain all four modes of a universal shift register (SISO, SIPO, PISO, PIPO), compare their timing diagrams, and describe practical applications.

p) Use Python to simulate shift register waveforms for each mode and compare automated output against hardware-observed waveforms.

Experiment 9 – MOD-8 Ripple Counter

q) Use AI/LLM tools to explain the modulus concept, the difference between synchronous and asynchronous (ripple) counters, and identify propagation delay issues in ripple counters.

r) Use Python to automate waveform generation for MOD-8 ripple counter states and compare simulated vs. observed waveforms programmatically.

Experiment 10 – MOD-8 Synchronous Counter

s) Use AI/LLM tools to explain the design procedure for a synchronous counter – state diagram, state table, flip flop excitation equations – and compare with the ripple counter approach.

t) Use AI-assisted state table generation to automatically derive T flip-flop excitation equations and validate the counter sequence.

Experiment 11 – Single-Bit Comparator

u) Use AI/LLM tools to derive the logic expressions for a 1-bit comparator ($A>B$, $A=B$, $A<B$), verify with truth table walkthrough, and generalise to N-bit comparator design.

v) Use Python to automate truth table generation and output verification for comparator circuits of arbitrary bit width.

Experiment 12 – BCD-to-7-Segment Display

w) Use AI/LLM tools to explain BCD-to-7-segment decoder logic, derive the segment activation truth table, and generate the segment mapping for all 10 BCD digits.

x) Build a Python-based AI tool that maps any BCD digit to the correct 7-segment display pattern using truth table lookup or logic minimisation.

Experiment 13 – BCD to Excess-3 Code Converter

y) Use AI/LLM tools to derive the logic equations for BCD-to-Excess-3 and Excess-3-to-BCD conversion, minimise them using K-maps, and verify the conversion for all valid inputs.

z) Use Python (SymPy) to automate logic simplification for both conversion directions and verify the circuit output against the expected code mapping.

Experiment 14 – Verilog HDL Decision-Making System

aa) Use AI/LLM tools to assist in Verilog code generation for the decision-making system, debug syntax and logical errors, and explain the RTL structure.

bb) Simulate the Verilog design using ModelSim or Vivado; optionally train a simple decision tree classifier to replicate the logic of the designed system.

Note: The suggested tasks can be modified based on the decision of the subject expert.

AI Integration	AI/ML integration is embedded across all 14 experiments in two tiers: (i) AI/LLM tools (ChatGPT, Claude.ai, Gemini) are used for truth table generation, K-map grouping, Boolean minimisation verification, timing diagram interpretation, Verilog code generation, and RTL debugging – bridging theory with hardware observation; (ii) Python/SymPy and Scikit-learn tools are applied for automated SOP/POS minimisation, MUX input mapping, flip flop next-state classification, counter waveform simulation, comparator truth table automation, 7-segment pattern lookup, and decision tree replication of Verilog logic. Any six of the fourteen AI/ML task pairs must be completed, at the subject expert's discretion.

Software Tools: ModelSim / Vivado (Verilog simulation); Python (SymPy, NumPy, Scikit-learn) – Google Colab (free); ChatGPT / Claude.ai / Gemini – LLM assistance; Digital Trainer Kit – hardware verification.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Design and verify combinational circuits including logic gates, full adder, comparator, decoder, multiplexer, and code converters by applying Boolean algebra, truth tables, and K-map minimisation techniques.	L3 (Apply)
CO2	Analyse and verify sequential circuits including SR, JK, and D flip flops, ring counters, shift registers, and MOD-8 ripple and synchronous counters, relating hardware behaviour to state diagrams, excitation tables, and timing diagrams.	L4 (Analyse)
CO3	Apply Karnaugh Map minimisation and excitation table methods to derive minimal logic expressions and flip flop input equations for the design of synchronous sequential circuits.	L4–L5 (Analyse, Evaluate)
CO4	Design and simulate digital systems using Verilog HDL in	L5 (Evaluate)

CO	Course Outcome Statement	Bloom's Taxonomy Level
	ModelSim or Vivado, verify RTL structure, and interpret simulation waveforms against expected functional behaviour.	
CO5	Use AI/LLM tools for truth table generation, logic derivation, and Verilog debugging, and Python/Scikit-learn models to automate Boolean minimisation, predict sequential circuit states, verify waveforms, and replicate digital logic using classifiers.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO5	State De Morgan's theorems; define SOP and POS; recall truth table of JK flip flop; define modulus of a counter; state the forbidden state of SR flip flop
L2 – Understand	Explain concepts, describe procedures	CO1–CO5	Explain NAND as universal gate; describe K-map grouping rules; explain race-around condition in JK flip flop; describe SISO vs. PIPO shift register operation; explain RTL structure of Verilog module
L3 – Apply	Design, implement, verify	CO1	Implement full adder using logic gates; verify decoder truth table; realise four-variable function using MUX; construct BCD-to-Excess-3 converter; test comparator for all input combinations
L4 – Analyse	Compare, verify, interpret	CO2, CO3	Analyse timing diagrams of flip flops; compare ripple vs. synchronous counter propagation delay; verify ring counter state sequence; derive T flip-flop excitation equations for MOD-8 counter
L5 – Evaluate	Assess, validate, simulate	CO3, CO4, CO5	Evaluate K-map minimised circuit against original truth table; validate Verilog simulation waveforms; assess Python classifier accuracy for next-state prediction; evaluate automated SOP minimisation output
L6 – Create	Design, build, automate	CO5	Write Verilog HDL decision-making system; build Python MUX mapping tool; develop flip flop next-state classifier; create automated waveform comparator for shift register modes; build 7-segment BCD pattern generator

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
CO1	L3	3	2	2	1	2	–	1	1	1	1	–	1	3
CO2	L4	3	3	2	1	2	1	1	1	2	1	–	2	3
CO3	L4-L5	3	3	2	1	2	1	1	1	2	2	–	2	3
CO4	L5	3	2	3	1	3	1	1	1	2	2	1	2	3
CO5	L5-L6	2	3	3	1	3	1	1	1	2	2	1	3	2

Recommended Tools & Platforms

Recommended Tools & Platforms
ModelSim / Vivado (Xilinx) – Verilog HDL simulation, RTL waveform verification, and synthesis for Experiment 14
Digital Trainer Kit – hardware verification of all combinational and sequential experiments (Experiments 1–13)
Python (SymPy, NumPy, Scikit-learn) on Google Colab (free) – Boolean minimisation, truth table automation, flip flop classifiers, counter waveform simulation
ChatGPT / Claude.ai / Gemini – LLM-based truth table generation, K-map assistance, Verilog code generation, and RTL debugging
LibreOffice Calc (free, open-source) – truth table and waveform data recording templates for all experiments

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI / ML Lab Exercises (any 6)	6 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs (CO1–CO5)
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI/ML exercises to be submitted as Google Colab .ipynb notebooks or ModelSim/Vivado simulation screenshots with Python scripts.

II Year I Sem

SUSTAINABILITY AND GREEN ENGINEERING

JNTUA R26 | II Year II Semester | All Branches

Course Code	Year/Sem	L	T	Pr	P	Credits
26AHS00305T	II Year II Sem	2	0	0	0	2

Prerequisites: Basic Environmental Science, Engineering Chemistry

Course Objectives

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

Course Content

UNIT I Fundamentals of Sustainability & Carbon Cycle

Core 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: CO_2, CH_4, N_2O, HFCs. Carbon dioxide equivalent (CO_2e). CO_2 contribution from cement production (~8% global emissions), steel, glass, aluminium. 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 quantities. Introduction to climate modelling data processing using Python (pandas, matplotlib). Online tools: Global Carbon Atlas, openLCA (free) for life cycle assessment.

Free/Open-Source AI Tools:

openLCA (free, open-source LCA software); Python (Pandas, Matplotlib) – Google Colab; Global Carbon Project tools; SimaPro (student trial)

UNIT II Sustainable Construction Materials

Core Content:

Construction materials and indoor air quality: VOCs, formaldehyde. No/low cement concrete: supplementary cementitious materials (fly ash, GGBS, silica fume, rice husk ash) — properties, mix proportions. Recycled aggregate concrete: sources, properties, durability. Alternative binders: alkali-activated materials (AAM), geopolymer concrete. Manufactured sand (M-sand): processing, quality. Bamboo, timber, and fibre-reinforced composites. Durability and service life assessment. Life cycle of construction materials.

AI Integration:

ML for sustainable material selection: multi-criteria decision analysis using Python. AI-based prediction of concrete strength from mix proportions using regression models (scikit-learn). Computer vision for recycled aggregate quality assessment. Introduction to materials informatics: using ML to discover new sustainable binders. Comparative LCA using openLCA.

Free/Open-Source AI Tools:

scikit-learn (free Python ML library); openLCA (free); Python (NumPy, scikit-learn) – Google Colab; Concrete Mixture Proportioner (free online NRMCA tool)

UNIT III Embodied Energy & Operational Energy in Buildings

Core Content:

Definition of embodied energy: extraction, processing, manufacturing, transport, construction, demolition. Operational energy: heating, cooling, lighting, hot water. Life cycle energy: sum of embodied and operational energy. Calculation of embodied energy for common construction materials (energy intensity values). Embodied energy calculation for a simple building structure. Energy balance: reducing operational energy vs. increasing insulation (embodied energy trade-off).

AI Integration:

Python-based embodied energy calculator for building elements. AI for building energy optimization: overview of EnergyPlus (free) and OpenStudio (free) simulations. ML for operational energy prediction from building parameters (Linear Regression, Random Forest using scikit-learn). Digital twin for energy monitoring: concept and tools. Building energy benchmarking using ML: comparison with national average.

Free/Open-Source AI Tools:

EnergyPlus (free, US DOE); OpenStudio (free); Python (scikit-learn, NumPy) on Google Colab (free); openLCA (free); AI tools: ChatGPT / Gemini

UNIT IV Green Building Rating Systems & Energy Codes

Core Content:

Energy Conservation and Sustainability Building Code (ECSBC-2023): scope, mandatory and prescriptive requirements, OTTV calculations, compliance path. Green building rating systems: LEED — credit categories, certification levels; GRIHA — India's national system; IGBC rating systems; BEE star rating. Passive design strategies: building orientation, natural ventilation, daylighting, cool roofs. Net Zero Energy Buildings (NZEB): definition, pathways.

AI Integration:

AI for automated LEED/GRIHA compliance checking: overview of existing tools. ML for building performance prediction: predicting LEED score from building design parameters. BIM-based energy analysis: linking Revit model to energy simulation. Smart building systems: AI-driven HVAC optimization. IoT sensors for real-time energy monitoring. Case study: AI-optimized green building design.

Free/Open-Source AI Tools:

EnergyPlus (free); OpenStudio (free); Autodesk Revit student version with Energy Analysis add-in (free); Python for data analysis (Google Colab)

UNIT V	Circular Economy, Waste Management & Environmental Sustainability

Core Content:

Circular economy principles: reduce, reuse, recycle, recover, redesign — waste hierarchy. Construction and demolition (C&D) waste: types, quantities, recycling potential. C&D waste management rules 2016. Deconstruction vs. demolition. Industrial waste in construction: fly ash, blast furnace slag, quarry dust, plastic waste utilisation. Water recycling in construction. Green procurement policies. Carbon sequestration potential of green infrastructure: urban trees, green roofs, wetlands. Acid rain: causes, effects, prevention.

AI Integration:

AI for construction waste prediction: ML models for waste generation estimation. Computer vision for C&D waste sorting (overview of automated sorting robots). Digital material passport: BIM-based end-of-life material tracking. ML for demolition waste characterization. Python for waste audit data analysis. Sustainable infrastructure planning using AI and GIS: overview of green infrastructure planning tools.

Free/Open-Source AI Tools:

Python (scikit-learn, Pandas) – Google Colab; QGIS (free GIS for green infrastructure planning); OpenLCA (free LCA); i-Tree (free urban tree carbon quantification tool by USDA)

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Remember and explain the principles of sustainability, SDGs, the carbon cycle, greenhouse gas accounting, and the environmental impact of civil engineering materials.	L1 (Remember) L2 (Understand)

CO	Course Outcome Statement	Bloom's Level
CO2	Apply knowledge of sustainable construction materials including SCMs, recycled aggregates, geopolymer concrete, and alternative binders to evaluate their environmental performance.	L3 (Apply)
CO3	Analyze the embodied energy and operational energy of building systems, perform energy calculations, and compare environmental performance of design alternatives.	L4 (Analyze)
CO4	Evaluate green building rating systems (LEED, GRIHA, ECBC), assess compliance with energy codes, and compare sustainability performance of different building design strategies.	L4 (Analyze) L5 (Evaluate)
CO5	Design sustainable civil engineering solutions integrating circular economy principles, C&D waste management, green infrastructure, and AI-driven optimization tools.	L5 (Evaluate) L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	2	1	–	–	–	3	3	2	–	2	–	2	2	2
CO2	L3	2	2	2	2	2	2	3	2	1	2	–	2	2	2
CO3	L4	2	2	2	2	3	2	3	1	1	2	1	2	2	3
CO4	L4,L5	2	2	3	2	3	3	3	2	1	2	1	3	2	3
CO5	L5,L6	2	3	3	3	3	3	3	2	2	3	2	3	2	3

Textbooks

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

Reference Books

1. ECSBC 2024 (Energy Conservation and Sustainability Building Code), Bureau of Energy Efficiency, Government of India.
2. Goswami D., Kreith F. and Kreider J., Principles of Solar Engineering, Taylor & Francis, 2000.

Web Resources & NPTEL Links

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

DATA STRUCTURES USING PYTHON

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26ACS05302S	II/ IYear	1	0	0	2	2	Skill Enhancement Course – I

Note: Any twelve experiments are to be performed.

Course Objectives

- Introduce fundamental data structures — linked lists, stacks, queues, trees, and graphs — and implement them using Python with practical coding exercises covering insertion, deletion, traversal, and search operations.
- Develop understanding of advanced tree-based structures including AVL Trees, Heaps, Tries, Segment Trees, and Fenwick Trees, with emphasis on self-balancing, priority management, prefix search, and range query efficiency.
- Familiarise students with graph algorithms — Dijkstra's, Bellman-Ford, Kruskal's, Prim's, and Topological Sort — and apply them to solve network optimisation and dependency resolution problems.
- Equip students with hashing techniques including collision resolution strategies (chaining, linear probing, quadratic probing, double hashing) and understand Python's built-in dict internals through hands-on experiments.
- Enable students to analyse time and space complexity of data structure operations, automate experimental data processing using Python, and develop AI-assisted applications integrating multiple data structures.

Unit-wise Syllabus with Experiments

UNIT I	Data Structures – Fundamentals

Topics / Concepts	Experiments
◦ Fundamentals of Python: Lists, Tuples, Dictionaries, Sets, Comprehensions, Functions, Recursion, Lambda expressions ◦ Linked Lists: Singly Linked List, Doubly Linked List, Circular Linked List — Insertion, Deletion, Traversal, Reversal, Merging two sorted lists ◦ Stacks: Stack using list and collections.deque, Applications — balanced parentheses, infix to postfix conversion, evaluation of postfix expressions ◦ Queues: Simple Queue, Circular	**1.** Implement a Singly Linked List with insert, delete, search, and reverse operations; display the list after each operation. **2.** Implement a Doubly Linked List; perform forward and backward traversal and deletion by value. **3.** Write a Python program using a Stack to check balanced parentheses, convert an infix expression to postfix, and evaluate the postfix expression. **4.** Implement a Circular Queue using a list; perform enqueue, dequeue, and display with overflow and underflow handling.

Queue, Deque (Double-ended Queue), Priority Queue using heapq module ◦ Trees: Binary Tree, Binary Search Tree (BST) — Insertion, Deletion, Traversals (Inorder, Preorder, Postorder) ◦ Graphs: Adjacency Matrix and Adjacency List representation, BFS and DFS traversals	**5.** Build a Binary Search Tree (BST) with insert, delete, and search operations; display Inorder, Preorder, and Postorder traversals and verify the BST property.

UNIT II	Advanced Data Structures – Trees and Heaps

Topics / Concepts ◦ AVL Trees: Self-balancing BST, Rotations — LL, RR, LR, RL, Insertion and Deletion with balancing factor ◦ Heaps: Min Heap and Max Heap, Heap operations using heapq, Heap Sort, Applications ◦ Tries: Trie construction, Insert, Search, and Delete operations, Applications in autocomplete and spell checking ◦ Segment Trees: Construction, Range query (sum, min, max), Point update, Lazy propagation ◦ Fenwick Tree: Construction, Prefix sum queries, Point updates, Applications	**Experiments** **1.** Implement AVL Tree insertion with all four rotations (LL, RR, LR, RL); display the balance factor and height after each insertion to confirm self-balancing. **2.** Implement Min Heap and Max Heap; perform insert, extract-min/max, and heapify operations and implement Heap Sort using Python's heapq module. **3.** Build a Trie data structure supporting insert, search, and delete of words; implement an autocomplete feature that suggests words for a given prefix. **4.** Build a Segment Tree for a given array; perform range sum and range minimum queries with point update support; verify correctness against brute-force results. **5.** Implement a Fenwick Tree (Binary Indexed Tree) to support prefix sum queries and point updates; compare query results with a naive prefix-sum array approach.

UNIT III	Advanced Data Structures – Graphs and Hashing

Topics / Concepts ◦ Graph Algorithms: Dijkstra's Shortest Path, Bellman-Ford Algorithm, Kruskal's and Prim's Minimum Spanning Tree,	**Experiments** **1.** Implement Dijkstra's algorithm using a priority queue to find the shortest path from a source vertex; display distances to all nodes. **2.** Implement Bellman-Ford algorithm on a

Topological Sort (Kahn's and DFS-based) ◦ Hashing: Hash functions, Collision resolution — Chaining and Open Addressing (Linear Probing, Quadratic Probing, Double Hashing), Python dict internals ◦ Sliding Window and Sparse Table: Sliding window maximum using deque, Sparse table for static Range Minimum Query (RMQ)	weighted directed graph; find shortest paths from a source vertex and detect negative weight cycles. **3.** Find the Minimum Spanning Tree of a weighted undirected graph using both Kruskal's algorithm (with Union-Find) and Prim's algorithm; compare the total MST cost. **4.** Implement Topological Sort using Kahn's BFS-based algorithm on a DAG and apply it to determine a valid execution order for a set of dependent tasks. **5.** Implement a Hash Table with insert, search, and delete using Separate Chaining and Open Addressing (Linear Probing); compare collision counts and performance for both methods.

AI Integration	AI integration adds three open-ended exercises following the regular experiments: (i) Complexity Visualiser (Python / Matplotlib) — instrument each data structure operation with time.perf_counter(); plot empirical time vs n curves and overlay theoretical Big-O bounds; identify crossover points between structures; (ii) Autocomplete Engine (Trie + ML ranking, Experiment Unit II–3) — build a Trie-based autocomplete system; rank suggestions using TF-IDF or a lightweight n-gram model trained on a sample corpus; evaluate precision@k; (iii) Graph Route Optimiser (NetworkX + Dijkstra's) — model a real city road network as a weighted graph; apply Dijkstra's and Bellman-Ford; visualise shortest paths using Matplotlib; compare runtime for sparse vs dense graphs — reinforcing the connection between theoretical algorithm design and real-world network optimisation.

Free / Open-Source AI Tools: Python (NumPy, Matplotlib, NetworkX, scikit-learn) – Google Colab (free); VisuAlgo – visualgo.net (free); Python Tutor – pythontutor.com (free); GeeksforGeeks IDE (free).

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Design and implement fundamental data structures — linked lists, stacks, queues, binary trees, BST, and graphs — in Python; trace insertion, deletion, traversal, and search operations; analyse and compare their time and space complexities using Big-O notation.	L3 (Apply)
CO2	Apply advanced tree-based data structures — AVL Trees, Heaps, Tries, Segment Trees, and Fenwick Trees — to solve computational problems	L3–L4 (Apply,

CO	Course Outcome Statement	Bloom's Taxonomy Level
	involving self-balancing, priority scheduling, prefix search, and efficient range queries; verify correctness against brute-force results.	Analyse)
CO3	Use graph algorithms (Dijkstra's, Bellman-Ford, Kruskal's, Prim's, Topological Sort) to solve network optimisation, shortest path, and dependency resolution problems; compare algorithm performance on sparse and dense graphs.	L4 (Analyse)
CO4	Implement hashing techniques with collision resolution strategies (separate chaining, linear probing, quadratic probing, double hashing); compare collision counts and lookup performance; apply sliding window and sparse table methods for range-based problems.	L4–L5 (Analyse, Evaluate)
CO5	Automate complexity analysis and visualise empirical vs theoretical Big-O curves using Python; develop an AI-assisted application (autocomplete engine or graph route optimiser) integrating at least two data structures; evaluate model and algorithm performance using standard metrics.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state rules	CO1–CO4	Define linked list, stack, queue, tree, graph; state Big-O notations for common operations; recall AVL rotation cases; define hashing, collision, and hash function properties
L2 – Understand	Explain structures, describe logic	CO1–CO5	Explain difference between Min Heap and Max Heap; describe Trie vs BST for prefix search; explain chaining vs open addressing; describe Kahn's topological sort; describe sliding window technique
L3 – Apply	Implement, code, execute	CO1, CO2	Code Singly Linked List with all operations; implement Circular Queue with overflow handling; build BST and verify traversals; implement AVL rotations; code Heap Sort using heapq
L4 – Analyse	Compare, verify, trace	CO2, CO3, CO4	Compare Kruskal's vs Prim's MST cost; trace Dijkstra's vs Bellman-Ford on the same graph; compare Segment Tree vs Fenwick Tree query times; analyse collision counts for chaining vs probing
L5 – Evaluate	Assess performance, benchmark	CO4, CO5	Evaluate empirical vs theoretical Big-O curves; assess autocomplete precision@k; benchmark Hash Table methods; evaluate Dijkstra's runtime on sparse vs dense

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
			graphs; assess AI model R^2 and RMSE
L6 – Create	Design application, propose solution	CO5	Design and build a Python autocomplete engine using Trie + ML ranking; develop a graph route-optimiser for a real network; propose a hybrid data structure for a given problem; write complexity analysis report

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 Com mun.	PO11 PrjMgmt	PO12 Lifelong Learn.	PSO1 CS Core	PSO2 Profnl Dev.
CO1	**L3**	3	2	2	2	3	1	–	1	2	2	–	2	3	2
CO2	**L3–L4**	3	3	3	2	3	1	–	1	2	2	–	2	3	2
CO3	**L4**	3	3	2	3	3	1	1	1	2	2	–	2	3	2
CO4	**L4–L5**	3	3	2	3	3	1	–	1	2	2	–	2	3	3
CO5	**L5–L6**	2	3	3	3	3	2	1	1	2	3	1	3	2	3

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, Matplotlib, NetworkX, scikit-learn) on Google Colab (free) – complexity visualisation, autocomplete ML ranking, graph route optimisation
VisuAlgo (free) – visualgo.net – interactive step-by-step animation of linked lists, BST, AVL, heaps, sorting, graphs, and hashing
Python Tutor (free) – pythontutor.com – visualise call stacks, recursion traces, and object references for linked list and tree operations
GeeksforGeeks IDE (free) – online Python compiler with a built-in DSA problem set for practice after each experiment
LeetCode / HackerRank (free tier) – competitive programming problems mapped to each unit for assessment and self-evaluation

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI Lab Exercises (open-ended)	3 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI exercises to be submitted as Google Colab .ipynb notebooks with output screenshots and complexity analysis.

Learning Resources

Reference Books

S.No	Title	Author(s)	Publisher	Year
1	Data Structures and Algorithms in Python	Goodrich, Tamassia& Goldwasser	Wiley	2013
2	Grokking Algorithms: An Illustrated Guide for Programmers and other Curious People	Aditya Y. Bhargava	Manning Publications	2016
3	Introduction to Algorithms, 4th Edition	Cormen, Leiserson, Rivest & Stein	MIT Press	2022

Online Learning Resources / Virtual Labs

- GeeksforGeeks – Data Structures Portal – https://www.geeksforgeeks.org/data-structures/
- VisuAlgo – Interactive Visualisation of Data Structures and Algorithms – https://visualgo.net/en
- Coursera – Data Structures and Algorithms Specialisation – https://www.coursera.org/specializations/data-structures-algorithms
- NPTEL – Programming, Data Structures and Algorithms using Python – https://nptel.ac.in
- Python Tutor – Visualise Python Execution – https://pythontutor.com

II Year I Sem

TECHNICAL REPORT WRITING & INTELLECTUAL PROPERTY RIGHTS

JNTUA R26 | II Year II Semester | Common to All Branches | Audit Course

Course Code	Year/Sem	L	T	Pr	P	Credits
26AHS00302M	II Year I Sem	2	0	0	0	0

Prerequisites: Communicative English

Course Overview

This audit course equips engineering students with professional technical writing competencies and a comprehensive understanding of Intellectual Property Rights. The course follows a building-block approach: foundational concepts of technical communication are introduced first, followed by progressive application in report writing, research documentation, and IP strategy. AI-powered writing tools are embedded throughout to enhance productivity while reinforcing principles of academic integrity.

Course Objectives

- Introduce students to technical communication principles and types of technical documents used in engineering practice.
- Develop skills in writing well-structured technical reports, research proposals, and scientific papers.
- Expose students to the Intellectual Property Rights system — patents, trademarks, copyrights, and trade secrets — and the patent filing process in India.
- Enable students to conduct patent searches, analyse patent claims, and draft a basic patent disclosure.
- Integrate AI writing tools throughout the course while building critical evaluation and academic integrity skills.

Course Content

UNIT I Foundations of Technical Communication

Core 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; adapting tone and technicality. Language in technical writing: active vs. passive voice, nominalisations, clarity and conciseness.

AI Integration:

Analyse a technical paragraph for passive voice overuse and nominalisation, Hemingway Editor – paste a technical section and improve readability score to Grade 10 or below, Compare two versions of a technical document (before and after AI editing) and document changes

Free/Open-Source AI Tools:

Grammarly (free); Hemingway Editor (free online); ChatGPT / Gemini (free tiers)

UNIT II Technical Report Writing

Core 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; 150-250 word limit. Literature review: searching, evaluating, and synthesising sources. Data presentation: tables, figures, charts — placement, numbering, and captioning. Citation and referencing: IEEE, APA, and MLA styles; avoiding plagiarism.

AI Integration:

AI-powered literature search; find and summarise 5 relevant papers, manage citations and auto-generate bibliographies, generate a first-draft abstract; critique and rewrite to meet technical writing standards, Use Turnitin (institutional access) or Copyleaks (free tier) to verify originality

Free/Open-Source AI Tools:

Elicit.org (free); Zotero (free, open-source); ChatGPT / Gemini (free tiers); Copyleaks (free tier) for plagiarism checking

UNIT III Research Proposals and Scientific Writing

Core Content:

Research proposal structure: background, problem statement, objectives, methodology, timeline, budget. Scientific paper format: IMRAD structure (Introduction, Methods, Results and Discussion). Writing results vs. discussion: factual reporting vs. interpretation. Peer review process: understanding blind review, revision, and resubmission. Conference papers vs. journal articles: scope, length, and submission guidelines.

AI Integration:

Search for open-access papers in your domain; analyse IMRAD structure, get AI feedback on scientific language in a draft methods section, write a 1-page research proposal for a small project; use AI for structural review and language polishing

Free/Open-Source AI Tools:

Semantic Scholar (free); Writefull (free); ChatGPT / Gemini (free tiers); Google Scholar (free)

UNIT IV Intellectual Property Rights – Fundamentals

Core 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, patent application process. Patentability criteria: novelty, inventive step (nonobviousness), industrial applicability. Patent search: prior art search using databases. Copyright in engineering: software copyright, open-source licenses (GPL, MIT, Apache).

AI Integration:

Conduct a prior art search on a given invention idea, search and analyse patent documents, identify 3 patents related to a chosen engineering domain and analyse their claims section, how does AI help in patent drafting?

Free/Open-Source AI Tools:

Google Patents (free); Lens.org (free); ChatGPT / Gemini for patent language explanation (free tiers)

UNIT V Patent Drafting and IP Strategy

Core Content:

Structure of a patent application: title, abstract, background, summary, detailed description, claims, drawings. Types of claims: independent and dependent claims; method, product, system 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:

Draft a set of patent claims for a simple mechanical invention; critique for clarity and scope, IP India Online Portal (ipindia.gov.in) – navigate and understand the e-filing system, write a complete patent disclosure document for a project-based invention idea, use AI to convert a project report executive summary into a patent abstract; compare the two

Free/Open-Source AI Tools:

Google Patents (free); Lens.org (free); IP India Online Portal (ipindia.gov.in — free); ChatGPT / Gemini (free tiers)

Course Outcomes (COs)

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 (IPR).	L1 (Remember) L2 (Understand)
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 (Analyze)
CO4	Evaluate patent documents, prior art, and IP claims to assess novelty and inventiveness of an engineering innovation.	L5 (Evaluate)

CO	Course Outcome Statement	Bloom's Level
CO5	Create a complete technical report and draft a patent disclosure for an engineering project, integrating AI tools for writing assistance.	L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	1	1	1	1	1	1	1	2	1	3	2	2	2	2
CO2	L3	2	2	2	2	2	1	1	2	2	3	2	3	2	2
CO3	L4	2	3	2	2	3	1	1	2	2	3	2	3	2	2
CO4	L5	2	3	2	3	2	2	2	3	2	2	2	3	3	3
CO5	L6	2	2	3	2	3	1	1	2	3	3	3	3	3	3

Textbooks

1. Markel M. and Selber S., Technical Communication, 13th Edition, Bedford/St. Martin's, New York, 2022.
2. Ahuja B.N. and Ahuja S.S., Intellectual Property Rights, 4th Edition, Khanna Publishers, New Delhi, 2021.

Reference Books

1. Lannon J. and Gurak L., Technical Communication, 15th Edition, Pearson Education, 2020.
2. Meganathan R., Intellectual Property Rights for Engineers, Universities Press (India), 2019.
3. WIPO, Intellectual Property Handbook, WIPO Geneva, 2022.

Web Resources

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

MANAGERIAL ECONOMICS AND FINANCIAL ANALYSIS

JNTUA R26 | II Year II Semester | Common to All Branches

Course Code	Year/Sem	L	T	Pr	P	Credits
26AMB02301T	II Year II Sem	2	0	0	0	2

Prerequisites: Basic Mathematics, Engineering Economics concepts

Course Objectives

- Understand 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 optimise organisational resource allocation.
- Analyse various forms of business organisations, 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 Content

UNIT I Introduction to Managerial Economics

Core 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:

Generate demand scenarios for an engineering product (e.g., solar panels) and interpret price elasticity, implement moving average and exponential smoothing for a sample sales dataset, use publicly available CMIE or RBI data; apply AI-assisted regression toforecast demand.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); Google Sheets + FORECAST function (free); Python (pandas, scikit-learn) on Google Colab (free); Wolfram Alpha (free)

UNIT II Production and Cost Analysis

Core 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; determination of Break-Even Point; margin of safety and profit planning (simple problems).

AI Integration:

Build a BEA spreadsheet with dynamic charts, generate a cost function for a manufacturing scenario and identify optimal production level, collect cost data of a local small enterprise; perform AI-assisted CVP analysis and present findings, visualise cost curves.

Free/Open-Source AI Tools:

Google Sheets (free); LibreOffice Calc (free); Python (Plotly, pandas) on Google Colab (free); ChatGPT / Gemini (free tiers); Wolfram Alpha (free)

UNIT III Business Organisations and Market Structures

Core Content:

Forms of Business Organisations — sole proprietary, partnership, joint stock companies, startups and digital enterprises. Public vs. private sector in digital economy. Market Structures — classification of markets with real-world examples. Pricing methods and strategies.

AI Integration:

Simulate a duopoly pricing game and find Nash Equilibrium with AI assistance, Analyse pricing strategies of three competing tech products using online data and AI classification, how do AI algorithms determine prices and dynamic pricing in e-commerce.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); Google Sheets (free); Python (pandas, Matplotlib) on Google Colab (free)

UNIT IV Capital Budgeting

Core Content:

Capital — introduction, meaning, nature, and significance. Capital Budgeting — concept, nature, significance in modern enterprises, role in strategic decision-making. Methods — Payback Method, Accounting Rate of Return (ARR), Net Present Value (NPV), Internal Rate of Return (IRR), Profitability Index (PI).

AI Integration:

Evaluate 3 capital investment alternatives, Monte Carlo simulation for capital budgeting risk analysis, AI-based working capital optimization using predictive models, decision tree analysis for investment appraisal.

Free/Open-Source AI Tools:

Google Sheets (free); Python (NumPy, Matplotlib, scikit-learn) on Google Colab (free); ChatGPT / Gemini (free tiers); Wolfram Alpha (free)

UNIT V Financial Accounting and Analysis

Core Content:

Accounting principles and digital accounting systems. Journal, ledger, trial balance. Final Accounts — trading account, profit and loss account, balance sheet with simple adjustments. Financial Analysis — liquidity ratios, activity ratios, capital structure ratios, profitability ratios.

AI Integration:

Automated financial statement, interpret a company's financial ratios from a pasted balance sheet and income statement, Machine learning for financial ratio prediction and company health assessment, download annual report of an Indian engineering company (BHEL, L&T, Infosys) and perform ratio analysis.

Free/Open-Source AI Tools:

Python (pandas, Matplotlib) on Google Colab (free); Google Sheets (free); ChatGPT / Gemini (free tiers); Wolfram Alpha (free)

Course Outcomes (COs)

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

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	1	2	1	1	1	2	1	1	1	2	2	2	1	1

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO2	L3	2	3	2	2	2	2	1	1	1	2	3	2	2	2
CO3	L4	2	3	2	3	3	2	1	1	2	2	3	3	2	2
CO4	L5	2	3	2	3	3	2	1	1	2	2	3	3	2	2
CO5	L6	2	3	3	3	3	2	2	1	3	3	3	3	3	3

Textbooks

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

Reference Books

1. Aryasri A.R., Managerial Economics and Financial Analysis, McGraw-Hill Education, Revised Edition, 2020.
2. Prasanna Chandra, Financial Management: Theory and Practice, Tata McGraw-Hill, 10th Edition, 2019.
3. Gupta G.S., Economics for Engineers, Tata McGraw-Hill Education, 2nd Edition, 2017.

Web Resources & NPTEL Links

- NPTEL: Managerial Economics – https://nptel.ac.in/courses/110105063
- NPTEL: Financial Accounting for Engineers – https://nptel.ac.in/courses/110107114
- Khan Academy – Microeconomics and Finance Basics – https://www.khanacademy.org
- SWAYAM Portal – Business Economics – https://swayam.gov.in

UNIVERSAL HUMAN VALUES – UNDERSTANDING HARMONY AND ETHICAL HUMAN CONDUCT

JNTUA R26 | II Year II Semester | Common to All Branches

Course Code	Year/Sem	L	T	Pr	P	Credits
26AHS00302T	II Year II Sem	3	0	0	0	3

Prerequisites: None

Course Overview

This course, recommended by AICTE and aligned with NBA guidelines, explores the foundational principles of human existence, values, and ethical conduct through a systematic, self-exploratory approach. It draws from Indian philosophical traditions and integrates contemporary perspectives including AI ethics. Students are encouraged to critically examine prevailing beliefs, use AI as a reflective tool, and develop a coherent value framework for their professional and personal lives.

Course Objectives

- Introduce students to the philosophy, principles, and stages of value education and its relevance to engineering practice.
- Enable students to explore harmony at individual, family, societal, and ecological levels through structured self-reflection.
- Develop an understanding of professional ethics, engineering codes of conduct, and the consequences of unethical practice.
- Expose students to the concept of ecological balance, sustainable development, and the engineer's responsibility to nature.
- Cultivate ethical reasoning through analysis of real engineering case studies, AI ethics debates, and personal value charters.

Course Content

UNIT I Introduction to Value Education and Human Consciousness

Core 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, Fulfilling. Difference between animal consciousness and human consciousness.

AI Integration:

Present ethical dilemmas to AI and compare AI responses with naturally accepted human values, use an AI journalling assistant to document daily natural acceptances and violations, can AI ever have human values? Debate using structured arguments

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); Notion AI (free tier) for journalling

UNIT II Harmony in the Individual

Core Content:

Harmony between Self and Body: needs of the Self (happiness, trust) vs. needs of the Body (food, shelter, clothing). Four dimensions of human being: 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: holistic definition beyond materialistic view. Self-regulation vs. external regulation: inner motivation and its role in ethical conduct.

AI Integration:

Explore conversations about emotional needs; critically reflect on AI limitations in understanding values, create a 'Well-being Wheel'; use AI to identify gaps between material and value-based happiness, write a weekly reflection on one feeling (e.g., trust) experienced and observed.

Free/Open-Source AI Tools:

Replika (free); Canva (free); Notion AI (free tier); ChatGPT / Gemini (free tiers)

UNIT III Harmony in the Family and Society

Core 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 — undivided society. Universal Human Order: five dimensions of human existence (Individual, Family, Society, Nature, Existence). Social justice: ensuring rights, duties, and dignified participation for all.

AI Integration:

Generate scenarios of family conflicts and analyse through the lens of trust and respect, use AI to map historical social movements (Gandhi, Ambedkar) onto the five-order framework, document 3 community service observations and analyse them using the harmony framework; use AI for structuring the report.

Free/Open-Source AI Tools:

ChatGPT / Gemini (free tiers); Notion AI (free tier) for report structuring

UNIT IV Harmony in Nature and Existence

Core 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:

Analyse deforestation data and correlate with lack of ecological values, generate a sustainable engineering project idea and critique it using ecological harmony principles, create an ecological audit of the campus using satellite images (Google Earth) and document AI-assisted analysis

Free/Open-Source AI Tools:

Global Forest Watch (free); Google Earth (free); ChatGPT / Gemini (free tiers)

UNIT V Ethical Human Conduct and Professional Ethics

Core 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 and universally prosperous world. AI Ethics: bias, fairness, transparency, and accountability in AI systems.

AI Integration:

Demonstrate AI bias using sample datasets, analyse Bhopal Gas Tragedy / Challenger Disaster through professional ethics lens, use a given AI system and evaluate it on fairness and transparency, write a 'Personal Value Charter' as a future engineer; use ChatGPT for drafting and self-editing for authenticity.

Free/Open-Source AI Tools:

IBM AI Fairness 360 (free, open-source); ChatGPT / Gemini (free tiers); Notion AI (free tier) for charter drafting

Course Outcomes (COs)

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 (Remember) L2 (Understand)
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 (Analyze)
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	L6 (Create)

CO	Course Outcome Statement	Bloom's Level
	dialogue.	

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	–	–	–	–	–	2	2	3	1	2	–	2	1	1
CO2	L3	–	–	–	–	–	3	2	3	2	2	1	2	1	1
CO3	L4	1	2	–	1	–	3	2	3	2	2	1	3	1	2
CO4	L5	1	2	2	2	2	3	3	3	2	2	2	3	2	2
CO5	L6	1	2	2	1	2	3	2	3	3	3	2	3	2	2

Textbooks

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

Reference Books

1. Subramanian C.V., Professional Ethics and Human Values, Oxford University Press, 2020.
2. Bajpai S.C., Engineering Ethics: Concepts and Cases, McGraw-Hill Education, 2019.
3. Covey S.R., The 7 Habits of Highly Effective People, Simon & Schuster, 30th Anniversary Edition, 2020.
4. Singer P., Practical Ethics, 4th Edition, Cambridge University Press, 2021.

Web Resources

- AICTE – Value Education Resources – https://aicte-india.org
- UNESCO – Ethics of AI – https://www.unesco.org
- IEEE Ethics Resources – https://ethics.iit.edu
- NPTEL – Human Values & Professional Ethics – https://swayam.gov.in

II Year II Sem

ELECTRONIC CIRCUITS ANALYSIS

JNTUA R26 | II Year II Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04401T	II Year II Sem	2	0	2	0	3

Prerequisites: Electronic Devices and Circuits, Network Analysis

Course Objectives

- Analyse multistage amplifiers, differential amplifiers, and understanding coupling methods.
- Understand frequency response of amplifiers and evaluate gain-bandwidth product.
- Analyse feedback amplifier topologies and design oscillators satisfying the Barkhausen criterion.
- Analyse power amplifiers in terms of efficiency, distortion, crossover characteristics, and thermal design of BJTs and MOS transistors.
- Design tuned amplifiers and multivibrators using transistors.

Course Content

UNIT I — Multistage & Differential Amplifiers

Core Content:

Introduction, classification of amplifiers, distortion in amplifiers, coupling schemes, cascaded RC-coupled BJT amplifiers, cascode amplifier, Darlington pair, current mirror, MOS differential pair, small-signal operation of the MOS differential pair, the BJT differential pair.

AI Integration:

Students use AI/LLM tools to understand RC coupling, cascode, and Darlington configurations, and to walk through CMRR calculation for a BJT differential amplifier. A 2-stage RC-coupled CE amplifier's gain is computed in Python (Google Colab) to confirm the multiplicative cascade relationship.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini

UNIT II — Frequency Response of Amplifiers

Core Content:

Low-frequency response of the CS and CE amplifiers, internal capacitive effects and the high-frequency model of the MOSFET and the BJT, high-frequency response of the CE, CS, CD amplifiers, f_β, f_T and gain-bandwidth product.

AI Integration:

Students use AI/LLM tools to understand f_β, f_T, and Miller's theorem for estimating high-frequency cutoff. The Bode magnitude response of a CE amplifier is plotted in Python (Google Colab) to measure mid-band gain, cutoff frequency, and bandwidth.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib, control) on Google Colab; ChatGPT / Gemini

UNIT III Feedback Amplifiers & Oscillators

Core Content:

Feedback Amplifiers: Introduction, the general feedback structure, properties of negative feedback, the four basic feedback topologies — series-shunt, series-series, shunt-shunt, shunt-series. Oscillators: General considerations, phase shift oscillator, Wien-bridge oscillator, LC oscillators, crystal oscillators, illustrative problems.

AI Integration:

Students use AI/LLM tools to identify feedback topologies, compute closed-loop gain, and verify the Barkhausen criterion for a phase-shift oscillator. Wien-bridge oscillator frequency is computed in Python (Google Colab) for varying R and C values to confirm $f = 1/(2\pi RC)$.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib, control) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini

UNIT IV Power Amplifiers

Core Content:

Introduction, Class A amplifiers (series fed, transformer coupled, push-pull), harmonic distortion, Class B amplifiers (push-pull, complementary symmetry), crossover distortion and Class AB operation, Class C amplifiers, power BJTs, MOS power transistors.

AI Integration:

Students use AI/LLM tools to compare Class A, B, AB, and C amplifiers on efficiency, distortion, and applications. Class A and Class B output waveforms are plotted in Python (Google Colab) to compute efficiency and observe crossover distortion.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Electronics lectures

UNIT V Tuned Amplifiers & Multivibrators

Core Content:

Tuned Amplifiers: Introduction, single tuned amplifiers — Q-factor, frequency response; double tuned amplifiers — Q-factor, frequency response; concept of stagger tuning. Multivibrators: Bistable, monostable, and astable multivibrators and Schmitt trigger operation using transistors.

AI Integration:

Students use AI/LLM tools to understand the $BW = f0/Q$ relationship and design a monostable multivibrator for a given pulse width. Single/double-tuned frequency responses and an astable multivibrator waveform are simulated in Python (Google Colab) to verify bandwidth, Q-factor, and timing relationships.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; Falstad Circuit Simulator; ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Analyse multistage BJT and MOS amplifiers, compute overall gain, and evaluate differential amplifier CMRR and non-ideal characteristics.	L4 (Analyze)
CO2	Determine the low-frequency and high-frequency response of CE and CS amplifiers and compute gain-bandwidth product (fβ, fT).	L4 (Analyze)
CO3	Apply negative feedback analysis to the four feedback topologies and design RC and LC oscillators satisfying the Barkhausen criterion.	L3 (Apply)
CO4	Evaluate efficiency, linearity, and thermal design considerations for Class A, B, AB, and C power amplifier configurations.	L5 (Evaluate)
CO5	Design single and double tuned amplifiers for specified Q-factor and bandwidth, and design astable, monostable, bistable multivibrators and Schmitt trigger circuits.	L6 (Create)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO2	L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO3	L3	3	3	3	2	2	–	–	–	1	2	–	1	3	2
CO4	L5	3	3	2	2	2	–	–	–	1	2	–	2	3	2
CO5	L6	3	3	3	2	3	–	–	–	1	2	1	2	3	3

Textbooks

1. Sedra A.S. and Smith K.C., Microelectronic Circuits, 6th Edition, Oxford University Press, 2011.
2. Millman J. and Halkias C., Integrated Electronics, 4th Edition, McGraw Hill Education (India), 2015.

Reference Books

1. Razavi B., Fundamentals of Microelectronics, Wiley, 2010.
2. Neamen D.A., Electronic Circuits – Analysis and Design, 3rd Edition, McGraw Hill, 2019.
3. Boylestad R.L. and Nashelsky L., Electronic Devices and Circuits Theory, 9th Edition, Pearson, 2006.

II Year II Sem

ANALOG AND DIGITAL COMMUNICATIONS

JNTUA R26 | II Year II Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04402T	II Year II Sem	2	0	2	0	3

Prerequisites: Signals, Systems and Stochastic Processes, Network Analysis

Course Objectives

- Understand the need for modulation and analyse Amplitude Modulation and demodulation methods.
- Understand FM and PM, analyse and study generation and detection with pre-emphasis/de-emphasis.
- Study AM/FM transmitter architectures, superheterodyne receiver characteristics and AM vs. FM receiver comparison.
- Understand Pulse modulation techniques and analyse them.
- Analyse coherent digital modulation schemes and study baseband transmission, probability of error, ISI, and eye diagrams.

Course Content

UNIT I Amplitude Modulation

Core Content:

Need for modulation. Amplitude Modulation — time and frequency domain description, single tone modulation, power relations in AM waves, generation using switching modulator, detection using envelope detector. DSB-SC modulation — time and frequency domain description, generation using balanced modulators, coherent detection. SSB modulation — time and frequency domain description, frequency discrimination and phase discrimination methods for generating SSB, demodulation of SSB waves. Principle of vestigial sideband modulation.

AI Integration:

Use AI/LLM tools to compare AM, DSB-SC, and SSB in a table (bandwidth, power efficiency, complexity) and solve modulation index and power problems. Plot AM waveforms in Python (Google Colab) for varying modulation index to observe envelope changes and overmodulation.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Communications lectures

UNIT II Angle Modulation

Core Content:

Basic concepts of phase modulation. Frequency Modulation: single tone FM, spectrum analysis using Bessel functions, narrow band FM, wide band FM, constant average power, transmission bandwidth of FM — Carson's rule. Generation of FM signal — Armstrong method. Detection of FM signal: phase locked loop. Comparison of FM and AM. Concept of pre-emphasis and de-emphasis.

AI Integration:

Use AI/LLM tools to verify Carson's rule (BW = 2(Δf + fm)), compare narrow-band and wide-band FM, and explain pre-emphasis/de-emphasis and PLL-based FM detection. Plot FM spectrum in Python (Google Colab) to observe Bessel function sidebands.

Free/Open-Source AI Tools:

Python (NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini

UNIT III Transmitters and Receivers

Core Content:

Classification of transmitters, AM transmitters, FM transmitters. Radio Receiver — receiver types, superheterodyne receiver, RF section and characteristics, frequency changing and tracking, intermediate frequency, image frequency, AGC, amplitude limiting, FM receiver, comparison of AM and FM receivers.

AI Integration:

Use AI/LLM tools to explain the superheterodyne receiver, solve IF and image frequency examples (fIF = fLO − fRF; fimage = fRF + 2·fIF), and compare AM and FM receivers in a table. Simulate the mixing stage in Python (Google Colab) to visualise IF extraction.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Communications lectures

UNIT IV Pulse Modulation

Core Content:

Types of pulse modulation — PAM, PWM and PPM. Comparison of FDM and TDM. Pulse Code Modulation: PCM generation and reconstruction, quantisation noise, non-uniform quantisation and companding, delta modulation, DPCM.

AI Integration:

Use AI/LLM tools to explain PAM, PWM, PPM, and PCM, solve a quantisation example (step size, SQNR), and compare FDM vs. TDM and μ-law/A-law companding. Plot original vs. quantised signals in Python (Google Colab) at 3-bit, 4-bit, and 8-bit to observe error reduction.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini

UNIT V Digital Modulation Techniques

Core Content:

Coherent digital modulation schemes — ASK, BPSK, BFSK, QPSK, DPSK. M-ary modulation techniques, power spectra, bandwidth efficiency. Baseband transmission and optimal reception of digital signals: baseband signal receiver, probability of error, optimum receiver, coherent reception, ISI, eye diagrams.

AI Integration:

Use AI/LLM tools to compare ASK, BPSK, BFSK, and QPSK in a table (bits/symbol, bandwidth, noise immunity) and explain ISI, eye diagram interpretation, and matched-filter receiver. Plot BPSK and QPSK signals in Python (Google Colab) to verify 90° phase spacing in QPSK constellation.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Digital Communications lectures.

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Analyse AM, DSB-SC, SSB, and VSB systems in time and frequency domains, determine power and bandwidth, and evaluate envelope detection, coherent detection, and balanced modulator methods.	L4 (Analyze)
CO2	Apply FM/PM principles to determine spectrum using Bessel functions, calculate bandwidth using Carson's rule, and evaluate Armstrong method and PLL-based detection.	L3 (Apply)
CO3	Evaluate AM/FM transmitter and superheterodyne receiver architectures, calculate IF and image frequency, and compare AM and FM receiver performance.	L5 (Evaluate)
CO4	Analyse PAM, PWM, PPM, PCM, delta modulation, and DPCM systems — computing quantisation noise, SQNR, companding advantage, and FDM/TDM trade-offs.	L4 (Analyze)
CO5	Apply ASK, BPSK, BFSK, QPSK, DPSK, and M-ary techniques, evaluate BER and bandwidth efficiency, and use LLMs and Python to simulate and analyse digital communication systems.	L3 (Apply) L4 (Analyze)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L4	3	3	2	2	2	–	–	–	–	2	–	1	3	2
CO2	L3	3	3	2	2	2	–	–	–	–	1	–	1	3	2
CO3	L5	3	3	2	2	2	–	–	–	1	2	–	2	3	2
CO4	L4	3	2	1	2	2	–	–	–	–	1	–	2	3	2
CO5	L3,L4	3	3	2	2	3	–	–	–	1	2	–	2	3	3

Textbooks

1. Haykin S., Communication Systems, 4th Edition, John Wiley & Sons, 2004.
2. Tomasi W., Electronics Communication Systems — Fundamentals through Advanced, 5th Edition, PHI, 2009.

Reference Books

1. Shanmugam S., Digital and Analog Communication Systems, John Wiley & Sons, 1999.
2. Sklar B. and Harris F.J., Digital Communications: Fundamentals and Applications, Pearson, 2020.
3. Taub H. and Schilling D.L., Principles of Communication Systems, Tata McGraw Hill, 2007.
4. Lathi B.P. and Ding Z., Modern Digital and Analog Communication Systems, Oxford Press, 2011.

II Year II Sem

LINEAR CONTROL SYSTEMS

JNTUA R26 | II Year II Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEE02401T	II Year II Sem	2	0	2	0	3

Prerequisites: Network Analysis, Signals and Systems, Engineering Mathematics

Course Objectives

- Understand the concepts of open-loop and closed-loop control systems, feedback, and derive mathematical models of mechanical and electrical systems.
- Analyse the time response of first-order and second-order control systems and design controllers to meet specified time-domain performance specifications.
- Apply the Routh-Hurwitz stability criterion and root locus technique to assess and predict closed-loop stability of linear control systems.
- Perform frequency-domain analysis of control systems to design compensators.
- Represent and analyse control systems in state space form and evaluate controllability and observability.

Course Content

UNIT I Control Systems Concepts & Mathematical Modelling

Core Content:

Open-loop and closed-loop control systems, examples, classification, feedback characteristics, effects of positive and negative feedback. Mathematical models — differential equations of translational, rotational mechanical and electrical systems, analogous systems, block diagram reduction, signal flow graphs, Mason's gain formula. DC servomotor and AC servomotor transfer functions, synchros.

AI Integration:

Use AI/LLM tools to apply Mason's gain formula to a signal flow graph and reduce a two-loop block diagram; verify manually. Plot the step response of a mass-spring-damper system in Python (Google Colab) to read ωn and damping ratio.

Free/Open-Source AI Tools:

Python (control, NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Control Systems lectures.

UNIT II Time Response Analysis

Core Content:

Step response, impulse response, time response of first-order systems, characteristic equation, transient response of second-order systems, time domain specifications, steady-state response, steady-state errors and error constants, P, PI, PD and PID controllers on second-order systems.

AI Integration:

Use AI/LLM tools to compute rise time, peak overshoot, and steady-state error using standard second-order formulas (ζ, ωn); verify manually. Plot under-damped, critically damped, and over-damped step responses in Python (Google Colab) and compare PD-controlled vs. uncontrolled responses.

Free/Open-Source AI Tools:

Python (control, NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini

UNIT III Stability Analysis in Time Domain

Core Content:

Concept of stability, Routh's stability criterion, stability and conditional stability, limitations of Routh's stability. Root locus concept, construction of root loci, effects of adding poles and zeros to G(s)H(s).

AI Integration:

Use AI/LLM tools to apply Routh-Hurwitz criterion to a 3rd-order polynomial, handle the row-of-zeros case, and verify the stability range of K for a second-order system. Plot the root locus in Python (Google Colab) to observe pole movement as gain K varies.

Free/Open-Source AI Tools:

Python (control, NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini

UNIT IV Frequency Response Analysis

Core Content:

Frequency domain specifications, Bode diagrams, transfer function from Bode diagram, stability analysis from Bode plots. Polar plots, Nyquist plots, phase margin and gain margin. Lag, lead, lag-lead compensator design in frequency domain.

AI Integration:

Use AI/LLM tools to read gain margin and phase margin from a Bode plot, apply the Nyquist criterion, and describe Lead compensator design steps with a numerical example. Plot Bode and Nyquist diagrams in Python (Google Colab) and compare gain/phase margins against analytical values.

Free/Open-Source AI Tools:

Python (control, NumPy, Matplotlib) on Google Colab; ChatGPT / Gemini; NPTEL Control Systems lectures.

UNIT V State Space Analysis of Continuous Systems

Core Content:

Concepts of state, state variables and state model, differential equations and transfer function models, block diagrams. Diagonalization, transfer function from state model, solving time-invariant state equations, state transition matrix and its properties. System response through state space, controllability and observability.

AI Integration:

Use AI/LLM tools to convert a transfer function to controllable canonical state space form, verify using $C(sI-A)^{-1}B+D$, and check controllability and observability via Kalman rank condition. Compute the state transition matrix in Python (Google Colab) and verify $\Phi(0) = I$.

Free/Open-Source AI Tools:

Python (control, NumPy, SciPy, Matplotlib) on Google Colab; ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Derive mathematical models, transfer functions, and block diagram representations of translational, rotational, and electrical control systems.	L3 (Apply)
CO2	Analyse transient and steady-state responses of first and second-order control systems and design PID controllers to meet time-domain specifications.	L4 (Analyze)
CO3	Apply Routh-Hurwitz criterion and construct root locus plots to determine stability and the effect of gain and pole-zero additions.	L3 (Apply) L4 (Analyze)
CO4	Construct Bode plots and Nyquist plots, determine gain and phase margins, and design Lag, Lead, and Lag-Lead frequency domain compensators.	L4 (Analyze) L5 (Evaluate)
CO5	Develop state space models, compute state transition matrices, and evaluate controllability and observability of continuous-time linear systems.	L4 (Analyze) L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	2	2	–	–	–	–	1	–	1	3	2
CO2	L4	3	3	3	2	2	–	–	–	–	1	–	2	3	2
CO3	L3,L4	3	3	2	2	2	–	–	–	–	1	–	1	3	2
CO4	L4,L5	3	3	3	2	3	–	–	–	1	2	–	2	3	2
CO5	L4,L5	3	3	3	2	2	–	–	–	–	1	–	2	3	2

Textbooks

1. Ogata K., Modern Control Engineering, 5th Edition, Prentice Hall of India, 2010.
2. Nagrath I.J. and Gopal M., Control Systems Engineering, 5th Edition, New Age International, 2007.

Reference Books

1. Gopal M., Control Systems Principles & Design, 4th Edition, McGraw Hill Education, 2012.
2. Kuo B.C. and Golnaraghi F., Automatic Control Systems, 8th Edition, John Wiley and Sons, 2003.
3. DiStefano J.J. III, Stubberud A.R., Williams I.J., Feedback and Control Systems, 2nd Edition, Schaum's Outlines, McGraw Hill, 2013.
4. Goodwin G.C., Graebe S.F., Salgado M.E., Control System Design, Pearson, 2000.

II Year II Sem

EM WAVES AND TRANSMISSION LINES

JNTUA R26 | II Year II Semester | ECE

Course Code	Year/Sem	L	T	Pr	P	Credits
26AEC04403T	II Year II Sem	2	0	2	0	3

Prerequisites: Physics for Communication Systems, Network Analysis, Engineering Mathematics (Vector Calculus)

Course Objectives

- Understand electrostatic field concepts and compute electric fields and capacitance for standard charge configurations.
- Apply Biot-Savart's law, Ampere's circuital law, and Faraday's law of induction to analyse magnetostatic fields and understand Maxwell's equations for time-varying fields.
- Derive and apply the electromagnetic wave equation, understand plane wave propagation in different media, and analyse reflection and refraction phenomena.
- Analyse transmission line parameters, derive the transmission line equations, and compute characteristic impedance and propagation constant for lossless and distortionless lines.
- Apply transmission line theory for input impedance, reflection coefficient, VSWR computation, and use the Smith chart for impedance matching and stub design.

Course Content

UNIT I — Fundamentals of Electrostatics

Core Content:

Coulomb's law, electric field intensity, electric fields due to continuous charge distributions, electric flux density, Gauss's law and applications, electric potential, Maxwell's two equations for electrostatic fields, energy density, convection and conduction currents, continuity equation and relaxation time, dielectric constant, Poisson's and Laplace's equations, capacitance — parallel plate, illustrative problems.

AI Integration:

Use AI/LLM tools to derive electric field and potential for point/line charges and verify field continuity at boundaries using Gauss's law. Plot field lines and equipotential contours in Python (Colab) by varying charge values to confirm the inverse-square law. Compute and compare capacitance for parallel-plate and coaxial geometries using Poisson's/Laplace's equations.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); PhET Electric Field simulation (free); AI tools: ChatGPT / Gemini; NPTEL EM Theory lectures

UNIT II — Magnetostatics & Maxwell's Equations

Core Content:

Biot-Savart law, Ampere's circuital law and applications, magnetic flux density, Maxwell's two equations for magnetostatic fields, magnetic scalar and vector potentials, forces due to magnetic fields, Ampere's force law, inductances and magnetic energy. Maxwell's Equations (Time Varying Fields): Faraday's law and transformer EMF, inconsistency of Ampere's law and displacement current density, Maxwell's equations in different final forms and word statements, conditions at a boundary surface.

AI Integration:

Use AI/LLM tools to explain displacement current and its role in correcting Ampere's law; state all four Maxwell's equations in integral form with physical interpretation. Plot magnetic field vs. distance for a long current-carrying conductor in Python (Colab) to confirm the 1/r dependence from Ampere's law.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); PhET Faraday's Law simulation (free); AI tools: ChatGPT / Gemini

UNIT III EM Wave Characteristics & Reflection/Refraction

Core Content:

Wave equations for conducting and perfect dielectric media, uniform plane waves — definition, relations between E and H, sinusoidal variations, wave propagation in lossy dielectrics, lossless dielectrics, free space, and good conductors, skin depth, polarisation and types. Reflection and Refraction of Plane Waves: Normal and oblique incidences for both perfect conductors and perfect dielectrics, Brewster angle, critical angle and total internal reflection, surface impedance, Poynting vector and Poynting theorem.

AI Integration:

Use AI/LLM tools to compute skin depth for copper at 1 GHz using $\delta = 1/\sqrt{(\pi f \mu \sigma)}$ and compare wave behaviour in free space, lossless dielectric, and good conductor. Plot plane wave amplitude vs. penetration depth in Python (Colab) varying conductivity; derive reflection/transmission coefficients and Brewster angle for a glass-air interface numerically.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); OpenEMS (free EM simulation); AI tools: ChatGPT / Gemini

UNIT IV Transmission Lines – I

Core Content:

Types, parameters, T and π equivalent circuits, transmission line equations, primary and secondary constants, expressions for characteristic impedance, propagation constant, phase and group velocities, infinite line, lossless lines, distortionless lines, illustrative problems.

AI Integration:

Use AI/LLM tools to derive $Z_0 = \sqrt{[(R+j\omega L)/(G+j\omega C)]}$ and simplify for lossless conditions; explain phase vs. group velocity and dispersion on a lossy line. Plot Z_0 and propagation constant vs. frequency in Python (Colab) to observe constant Z_0 and near-zero attenuation under lossless conditions.

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); AI tools: ChatGPT / Gemini; NPTEL Transmission Lines lectures

UNIT V Transmission Lines – II

Core Content:

Input impedance relations, reflection coefficient, VSWR, average power, shorted lines, open-circuited lines, and matched lines; low-loss radio frequency and UHF transmission lines. Smith chart — construction and applications, quarter-wave transformer, single stub matching, illustrative problems.

AI Integration:

Use AI/LLM tools to compute input impedance, reflection coefficient, and VSWR for a λ/4 shorted stub using $Z_{in} = jZ_0 \tan(\beta l)$; explain Smith chart construction for single-stub matching. Plot |Γ| and VSWR vs. normalised load impedance in Python (Colab) to identify the matched condition (Γ = 0, VSWR = 1).

Free/Open-Source AI Tools:

Python (NumPy, Matplotlib) on Google Colab (free); Smith chart calculators online (free); AI tools: ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Apply Coulomb's law, Gauss's law, and Poisson's/Laplace's equations to compute electric field intensity, potential, and capacitance for various charge distributions.	L3 (Apply)
CO2	Apply Biot-Savart's law, Ampere's law, and Maxwell's equations to analyse magnetostatic fields, time-varying fields, and electromagnetic boundary conditions.	L3 (Apply) L4 (Analyze)
CO3	Analyse uniform plane wave propagation in different media, compute attenuation and phase constants, skin depth, and evaluate reflection/refraction phenomena.	L4 (Analyze)
CO4	Compute transmission line parameters, characteristic impedance, and propagation constant; analyse lossless and distortionless line conditions.	L3 (Apply) L4 (Analyze)
CO5	Determine input impedance, reflection coefficient, and VSWR for terminated transmission lines; apply the Smith chart for single-stub impedance matching design.	L4 (Analyze) L5 (Evaluate)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	3	2	2	1	–	–	–	–	1	–	1	3	2
CO2	L3,L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO3	L4	3	3	2	2	2	–	–	–	–	1	–	2	3	2
CO4	L3,L4	3	3	2	2	2	–	–	–	–	1	–	1	3	2
CO5	L4,L5	3	3	3	2	3	–	–	–	1	2	–	2	3	3

Textbooks

1. Sadiku M.N.O., Elements of Electromagnetics, 7th Edition, Oxford University Press, 2008.
2. Jordan E.C. and Balmain K.G., Electromagnetic Waves and Radiating Systems, 2nd Edition, PHI, 2000.

Reference Books

1. Raju G.S.N., Electromagnetic Field Theory and Transmission Lines, 2nd Edition, Pearson Education, 2013.
2. Hayt W.H. Jr. and Buck J.A., Engineering Electromagnetics, 7th Edition, Tata McGraw Hill, 2006.
3. Krauss J.D., Electromagnetics, 3rd Edition, McGraw Hill, 1988.
4. Ryder J.D., Networks, Lines, and Fields, 2nd Edition, PHI Publications, 2012.

ELECTRONIC CIRCUITS ANALYSIS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04401P	II Year II Sem	0	0	0	3	**1.5**	Programme Core Lab – ECE

Note: Implement / execute any 12 experiments from the list below.

Course Objectives

- Introduce advanced amplifier configurations — Darlington pair, CE-CC multistage, cascode, differential, and feedback amplifiers — and determine their gain, bandwidth, and impedance characteristics.
- Develop understanding of RC/LC oscillators, Class A/AB power amplifiers, single-tuned amplifiers, and multivibrator circuits through hands-on design and analysis.
- Familiarise students with Schmitt trigger hysteresis, threshold voltage determination, and the impact of feedback on amplifier stability and bandwidth.
- Enable students to use AI/LLM tools to derive circuit equations, explain operation, and predict performance metrics such as gain, frequency, CMRR, efficiency, and pulse width.
- Develop ML regression and classifier models in Python/MATLAB to predict circuit performance parameters and classify waveform types from simulation or measured data.

List of Experiments

Part A – Advanced Amplifiers

1. Design and analyse the Darlington pair amplifier circuit. Determine the voltage gain, input impedance, and output impedance.
2. Plot and analyse the frequency response of a CE–CC (Common Emitter–Common Collector) multistage amplifier and determine bandwidth.
3. Design and analyse the Cascode amplifier circuit and determine its voltage gain, frequency response, and bandwidth.
4. Plot and analyse the frequency response of a Differential Amplifier and determine its CMRR (Common Mode Rejection Ratio).
5. Design and analyse any two topologies of feedback amplifiers (series-series, series-shunt, shunt-series, shunt-shunt) and find the frequency response of each.

Part B – Power Amplifiers and Oscillators

6. Design and analyse a Class A power amplifier. Determine efficiency, output power, and distortion.
7. Design and analyse a Class AB amplifier. Observe crossover distortion and compare performance with Class A.
8. Design and analyse an RC phase shift oscillator. Verify the oscillation frequency and observe the output waveform.
9. Design and analyse an LC oscillator (Colpitts or Hartley). Verify the oscillation frequency and observe the output waveform.

10. Plot the frequency response of a Single Tuned amplifier and determine its bandwidth and selectivity.

Part C – Multivibrators and Schmitt Trigger

11. Design a Bistable Multivibrator, analyse the effect of commutating capacitors, and draw the waveforms at the base and collector of transistors.
12. Design an Astable Multivibrator and draw the waveforms at the base and collector of transistors. Determine the oscillation frequency.
13. Design a Monostable Multivibrator and draw the input and output waveforms. Determine the pulse width.
14. Draw the response of a Schmitt trigger for gain greater than one and less than one. Determine the upper and lower threshold voltages.

AI / LLM and Python / MATLAB Tasks

Any six experiments need to be implemented using appropriate AI/LLM tools and/or Python/MATLAB-based simulations.

Exp. No.	Hardware / Simulation Experiment	AI / LLM & ML Task
1	Design and analyse the Darlington pair amplifier. Determine voltage gain, input impedance, and output impedance.	(a) Use AI/LLM to derive the Darlington pair current gain ($\beta\_D \approx \beta^2$), explain high input impedance advantage, and predict gain for different β values. (b) Train a regression model to predict overall voltage gain and input impedance from individual transistor parameters (β, r_e).
2	Plot and analyse the frequency response of a CE–CC multistage amplifier. Determine bandwidth.	(a) Use AI/LLM to explain the role of each stage (CE for gain, CC for impedance matching), predict the combined frequency response, and describe bandwidth shrinkage in cascaded stages. (b) Apply polynomial regression to model measured frequency response and automatically determine −3 dB bandwidth and midband gain.
3	Design and analyse the Cascode amplifier. Determine voltage gain, frequency response, and bandwidth.	(a) Use AI/LLM to explain the cascode configuration's advantage in reducing Miller effect, improving bandwidth, and achieving high gain; predict its frequency response. (b) Train an ML regression model to estimate voltage gain and bandwidth from component values and transistor parameters.
4	Plot and analyse the frequency response of a Differential Amplifier. Determine CMRR.	(a) Use AI/LLM to explain differential amplifier operation, define CMRR, and describe how the tail current source improves common-mode rejection; predict CMRR variation with circuit parameters.

Exp. No.	Hardware / Simulation Experiment	AI / LLM & ML Task
		(b) Build a regression model to predict CMRR from transistor parameters and resistor values.
5	Design and analyse any two feedback amplifier topologies and find frequency response of each.	(a) Use AI/LLM to compare the four feedback topologies (series-series, series-shunt, shunt-series, shunt-shunt); explain their effect on gain, impedance, bandwidth, and stability. (b) Train a classifier to identify the feedback topology from measured gain and impedance data; use regression to predict stability margins from open-loop parameters.
6	Design and analyse a Class A power amplifier. Determine efficiency, output power, and distortion.	(a) Use AI/LLM to explain Class A operation, DC load line and Q-point, and derive maximum efficiency expressions (25% resistive, 50% transformer-coupled). (b) Train a regression model to predict Class A efficiency and output power from Q-point location, supply voltage, and load resistance.
7	Design and analyse a Class AB amplifier. Observe crossover distortion and compare with Class A.	(a) Use AI/LLM to explain crossover distortion in Class B, how Class AB bias eliminates it, and compare efficiency and linearity trade-offs between Class A, B, and AB. (b) Develop an AI-based waveform classifier to detect and quantify crossover distortion from simulated or measured Class AB output signal data.
8	Design and analyse an RC phase shift oscillator. Verify oscillation frequency and waveform.	(a) Use AI/LLM to derive the RC phase shift oscillator frequency formula ($f = 1/2\pi\sqrt{6}\ RC$), explain the 180° phase shift condition, and predict frequency change with component variation. (b) Train a regression model to predict oscillation frequency from RC component values and verify against simulated/measured output.
9	Design and analyse a Colpitts or Hartley LC oscillator. Verify oscillation frequency and waveform.	(a) Use AI/LLM to explain the LC tank circuit, Barkhausen criterion, and derive the oscillation frequency from inductor and capacitor values. (b) Build a regression model to predict oscillation frequency from L and C parameter values and compare with the theoretical formula output.
10	Plot frequency response of a Single Tuned amplifier.	(a) Use AI/LLM to explain single-tuned amplifier operation, the role of the parallel LC tank circuit, and

Exp. No.	Hardware / Simulation Experiment	AI / LLM & ML Task
	Determine bandwidth and selectivity.	predict bandwidth and Q factor from circuit parameters. (b) Apply ML regression to estimate bandwidth and centre frequency from measured frequency response data points.
11	Design a Bistable Multivibrator. Analyse commutating capacitor effect and draw waveforms.	(a) Use AI/LLM to explain the two stable states of the bistable multivibrator, the role of commutating capacitors in improving switching speed, and predict the effect of capacitor value on switching time. (b) Train a waveform classifier to identify and label stable states and transition edges in bistable multivibrator waveforms.
12	Design an Astable Multivibrator. Draw waveforms and determine oscillation frequency.	(a) Use AI/LLM to derive the astable oscillation frequency ($f \approx 1/1.4RC$), explain charging/discharging cycles, and predict frequency change with component variation. (b) Train a regression model to predict oscillation frequency and duty cycle from RC timing component values.
13	Design a Monostable Multivibrator. Draw waveforms and determine pulse width.	(a) Use AI/LLM to explain the quasi-stable state, triggering mechanism, and derive pulse width expression ($T \approx 0.69RC$); predict pulse width for different RC values. (b) Build an ML pulse width prediction model (regression) that estimates output pulse width from RC component values and supply voltage.
14	Draw Schmitt trigger response for gain > 1 and < 1. Determine UTP and LTP.	(a) Use AI/LLM to explain Schmitt trigger hysteresis, derive UTP and LTP expressions, and explain behaviour for gain greater and less than unity. (b) Train a regression model to predict UTP and LTP threshold voltages from resistor ratio and supply voltage parameters.

Note: The suggested AI/ML tasks can be modified based on the decision of the subject expert.

AI Integration	AI integration adds six mandatory exercises alongside the regular hardware experiments. Students use LLM tools (ChatGPT / Claude / Gemini) to derive circuit equations and predict performance metrics, then validate predictions against measured values. ML tasks include: (i) regression models to predict voltage gain, CMRR, bandwidth, oscillation frequency, efficiency, and

	threshold voltages from component parameters using scikit-learn; (ii) waveform classifiers to detect crossover distortion in Class AB output and to label stable states in bistable multivibrator waveforms; (iii) feedback topology classifier trained on gain and impedance data to identify series/shunt configurations. All Python notebooks to be submitted as Google Colab .ipynb files with output plots and model evaluation metrics.

Free / Open-Source AI Tools: Python (NumPy, Matplotlib, scikit-learn, SciPy) – Google Colab (free); LTspice (free) – circuit simulation; MATLAB / GNU Octave (free) – frequency response analysis; ChatGPT / Claude (free tier) – LLM-based equation derivation and explanation.

Course Outcomes (COs)

After successful completion of this lab, students will be able to:

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Design and characterise advanced amplifier circuits — Darlington pair, CE-CC multistage, cascode, differential, and feedback amplifiers — measuring voltage gain, CMRR, input/output impedance, bandwidth, and frequency response using simulation and hardware.	L3 (Apply)
CO2	Analyse oscillator circuits (RC phase shift, LC Colpitts/Hartley, single-tuned) and verify the oscillation frequency using the Barkhausen criterion; determine bandwidth and selectivity (Q factor) from frequency response plots.	L3–L4 (Apply, Analyse)
CO3	Evaluate Class A and Class AB power amplifier performance by determining efficiency, output power, and distortion; observe and explain crossover distortion and compare Class A, B, and AB trade-offs.	L4 (Analyse)
CO4	Design bistable, astable, and monostable multivibrators and Schmitt trigger circuits; draw and interpret output waveforms; determine oscillation frequency, pulse width, duty cycle, and upper/lower threshold voltages.	L4–L5 (Analyse, Evaluate)
CO5	Apply ML regression models to predict circuit performance metrics (gain, CMRR, bandwidth, oscillation frequency, efficiency, threshold voltages) and develop waveform classifiers to detect distortion and identify multivibrator states from simulated or measured data.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	State Barkhausen criterion; define CMRR; recall Class A maximum efficiency (25% / 50%); define UTP and LTP; state Darlington pair $\beta_D \approx \beta^2$
L2 – Understand	Explain circuits, describe operation	CO1–CO5	Explain Miller effect and how cascode reduces it; describe crossover distortion in Class B; explain hysteresis in Schmitt trigger; describe role of commutating capacitors in bistable switching speed
L3 – Apply	Design, measure, conduct experiment	CO1, CO2	Design Darlington pair and measure gain, Z_in, Z_out; design RC phase shift oscillator and verify frequency; plot frequency response of single-tuned amplifier; design CE-CC amplifier and measure bandwidth
L4 – Analyse	Compare, verify, interpret waveforms	CO2, CO3, CO4	Compare Class A vs AB efficiency and distortion; verify oscillation frequency against theoretical formula; analyse commutating capacitor effect on bistable switching; interpret Schmitt trigger hysteresis loop
L5 – Evaluate	Assess accuracy, evaluate models	CO4, CO5	Evaluate regression model R^2 and RMSE for gain/CMRR prediction; assess waveform classifier accuracy for crossover distortion detection; evaluate pulse width predictor against measured monostable output
L6 – Create	Build ML models, develop applications	CO5	Build regression models for gain, CMRR, bandwidth, frequency, efficiency, UTP/LTP; develop crossover distortion classifier; create bistable waveform labeller; write complete Python analysis notebook

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 Investi gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Com mun.	PO11 PrjMgmt	PO12 Lifelong Learn.	PSO1 ECE Core	PSO2 Profnl Dev.
CO1	L3	3	2	3	3	3	1	–	1	2	2	–	1	3	2
CO2	L3–L4	3	3	3	3	3	1	–	1	2	2	–	2	3	2
CO3	L4	3	2	2	3	3	1	1	1	2	2	–	1	3	2
CO4	L4–L5	3	3	3	3	3	1	1	1	2	2	1	2	3	3
CO5	L5–L6	2	3	3	3	3	2	1	1	2	3	1	3	2	3

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, Matplotlib, SciPy, scikit-learn, pandas) on Google Colab (free) – regression models, waveform classifiers, frequency response plotting, and complexity visualisation
LTspice (free) – analog.com – SPICE-based circuit simulation for all 14 experiments; export waveform data directly to Python for ML training
MATLAB / GNU Octave (free) – frequency response and Bode plot analysis; polynomial fitting for −3 dB bandwidth determination
ChatGPT / Claude / Gemini (free tier) – LLM tools for equation derivation, circuit operation explanation, and performance prediction across all AI tasks
Virtual Labs (VLab) IIT – https://vlab.co.in – free online simulation of amplifier and oscillator experiments

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI / ML Lab Exercises	6 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI/ML exercises to be submitted as Google Colab .ipynb notebooks with simulation plots, model training code, and evaluation metrics (R^2, RMSE, accuracy).

Learning Resources

Text Books

S.No	Title	Author(s)	Publisher	Year
1	Electronic Devices and Circuit Theory, 12th Edition	Robert L. Boylestad& Louis Nashelsky	Pearson Education	2022
2	Integrated Electronics: Analog and Digital Circuits and Systems, 2nd Edition	Millman & Halkias	McGraw-Hill	2017
3	Electronic Circuits: Discrete and Integrated, 4th Edition	Donald L. Schilling & Charles Belove	McGraw-Hill	2019

Reference Books

S.No	Title	Author(s)	Publisher	Year
1	Analysis and Design of Analog Integrated Circuits, 5th Edition	Gray, Hurst, Lewis & Meyer	Wiley	2009
2	Microelectronics Circuit Analysis and Design, 4th Edition	Donald A. Neamen	McGraw-Hill	2010

Online Resources / Virtual Labs

1. NPTEL – Analog Electronic Circuits – https://nptel.ac.in
2. Virtual Labs IIT – https://vlab.co.in – amplifier and oscillator simulation experiments
3. LTspice Tutorials – https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html
4. MIT OpenCourseWare – 6.002 Circuits and Electronics – https://ocw.mit.edu
5. All About Circuits – https://www.allaboutcircuits.com – circuit theory reference and textbook

ANALOG AND DIGITAL COMMUNICATIONS LAB

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04402P	II Year II Sem	0	0	0	3	1.5	Programme Core Lab – ECE

Prerequisites: Analog Electronics; Signals & Systems; concurrent: Analog & Digital Communications (26A04401)

Course Objectives

- Introduce analog modulation schemes — AM, DSB-SC, FM, PAM, PWM, PPM — and FDM; include measurement of modulation index, frequency deviation, and channel bandwidth.
- Develop practical understanding of digital communication systems covering sampling theorem, TDM, Delta Modulation, PCM, and pulse modulation techniques.
- Implement digital modulation schemes — BPSK, BFSK, QPSK, DPSK — and evaluate system performance through BER vs. SNR curves, constellation diagrams, and SQNR measurements.
- Familiarise students with radio receiver characteristics including sensitivity, selectivity, and noise figure, and evaluate superheterodyne receiver architecture.
- Apply AI/LLM tools and Python/MATLAB-based simulations to derive modulation expressions, predict BER performance, detect aliasing, and classify modulation symbols using ML models.

List of Experiments

Design the circuits and verify the following experiments — minimum six from each section.

Section A — Analog Communication (Regular Experiments)

1. Design and implement AM modulation and demodulation circuits. Observe the modulated waveform, measure modulation index, and verify demodulation using envelope detection.
2. Design and implement DSB-SC (Double Sideband Suppressed Carrier) modulation and demodulation circuits and verify coherent detection.
3. Implement Frequency Division Multiplexing (FDM) for two or more signals and verify channel separation and bandwidth allocation.
4. Design and implement FM modulation and demodulation circuits. Measure the frequency deviation and modulation index.
5. Perform measurements on a radio receiver and evaluate sensitivity, selectivity, and noise figure.
6. Implement Pulse Amplitude Modulation (PAM) and demodulation. Observe the PAM waveform and verify signal recovery.

7. Implement Pulse Width Modulation (PWM) and demodulation. Observe the variation of pulse width with message signal amplitude.
8. Implement Pulse Position Modulation (PPM) and demodulation. Observe pulse position variation and verify signal recovery.

Section B — Digital Communication (Regular Experiments)

9. Verify the Sampling Theorem by sampling a continuous-time signal at different rates. Demonstrate aliasing when sampled below the Nyquist rate and perfect reconstruction above it.
10. Implement Time Division Multiplexing (TDM) for multiple signals and verify slot allocation and signal recovery.
11. Implement Delta Modulation and Demodulation. Observe granular noise and slope overload distortion and determine the optimal step size.
12. Implement Pulse Code Modulation (PCM) and demodulation. Observe quantisation error and measure signal-to-quantisation-noise ratio (SQNR).
13. Implement Binary Phase Shift Keying (BPSK) modulation and demodulation. Measure Bit Error Rate (BER) and plot the BER vs. SNR curve.
14. Implement Binary Frequency Shift Keying (BFSK) modulation and demodulation. Observe frequency switching and measure BER.
15. Implement Quadrature Phase Shift Keying (QPSK) modulation and demodulation. Plot the constellation diagram and measure BER.
16. Implement Differential Phase Shift Keying (DPSK) modulation and demodulation. Observe differential encoding and verify phase change detection.

Part B — AI/LLM and Python / MATLAB Integration Tasks

Any three experiments from each section to be implemented using AI/LLM tools and/or Python/MATLAB-based simulations.

Section A — Analog Communication: AI/LLM and Python/MATLAB Tasks

1. **Exp 1 (AM):** (a) Use AI/LLM tools to explain AM modulation theory, derive the modulation index expression, explain over-modulation, and describe envelope detector operation for demodulation. (b) Simulate AM modulation and demodulation in MATLAB/Python; apply regression to estimate modulation index from the measured waveform envelope data.
2. **Exp 2 (DSB-SC):** (a) Use AI/LLM tools to explain the suppressed carrier concept, the need for coherent detection, and how a product detector recovers the original signal. (b) Simulate DSB-SC modulation and coherent detection; use a regression or signal reconstruction model to analyse detection accuracy under different SNR conditions.
3. **Exp 3 (FDM):** (a) Use AI/LLM tools to explain the FDM concept, the role of guard bands, and how channel separation is achieved in the frequency domain; predict bandwidth requirements for N channels. (b) Simulate FDM with multiple channels and visualise channel separation; apply ML-based spectral analysis to verify channel isolation.

4. **Exp 4 (FM):** (a) Use AI/LLM tools to explain FM modulation principle, Carson's rule for bandwidth estimation, and the FM demodulation process using a discriminator or PLL. (b) Train a regression model to predict frequency deviation and bandwidth from modulation index and message signal frequency parameters.
5. **Exp 5 (Radio Receiver):** (a) Use AI/LLM tools to define receiver sensitivity, selectivity, and noise figure; explain the superheterodyne receiver architecture and predict performance trade-offs. (b) Use Python to model and simulate key receiver parameters; apply regression analysis to predict noise figure and sensitivity from circuit component values.
6. **Exp 6 (PAM):** (a) Use AI/LLM tools to explain PAM generation, the relationship between sampling frequency and signal recovery, and the effect of sampling rate on waveform quality. (b) Develop a Python-based waveform classifier to identify and categorise PAM signals from sampled waveform features and verify demodulation accuracy.
7. **Exp 7 (PWM):** (a) Use AI/LLM tools to explain PWM generation from comparator circuits, the relationship between message amplitude and pulse width, and the low-pass filter demodulation approach. (b) Build an ML regression model to predict output pulse width from the instantaneous message signal amplitude and verify against simulated PWM waveform data.
8. **Exp 8 (PPM):** (a) Use AI/LLM tools to explain PPM generation, how pulse position encodes information, and compare PPM with PAM and PWM in terms of noise immunity and bandwidth. (b) Train a classification model to detect and decode pulse position from PPM waveform data, mapping position values back to the original message signal.

Section B — Digital Communication: AI/LLM and Python/MATLAB Tasks

9. **Exp 1 (Sampling Theorem):** (a) Use AI/LLM tools to explain the Nyquist sampling theorem, conditions for aliasing and perfect reconstruction, and the relationship between sampling frequency and signal bandwidth. (b) Build a binary classifier (Logistic Regression) to detect aliasing from sampling rate and signal frequency parameters; simulate in Python/MATLAB and verify results.
10. **Exp 2 (TDM):** (a) Use AI/LLM tools to explain TDM frame structure, slot allocation, synchronisation requirements, and compare TDM with FDM in terms of efficiency and complexity. (b) Simulate TDM with multiple channels and visualise time-slot allocation; apply ML-based classification to verify correct channel recovery from the multiplexed signal.
11. **Exp 3 (Delta Modulation):** (a) Use AI/LLM tools to explain delta modulation operation, the causes of granular noise and slope overload, and the conditions for optimal step size selection. (b) Train a regression model to predict the optimal step size minimising total distortion (granular noise + slope overload) from message signal frequency and amplitude parameters.
12. **Exp 4 (PCM):** (a) Use AI/LLM tools to explain the PCM process (sampling, quantisation, encoding), derive the SQNR expression for uniform quantisation, and explain companding for non-uniform quantisation. (b) Build a Python-based quantisation error analyser; use regression to predict SQNR from the number of quantisation bits and input signal amplitude.
13. **Exp 5 (BPSK):** (a) Use AI/LLM tools to explain BPSK modulation/demodulation theory, derive the BER expression in AWGN, and explain coherent detection using a

matched filter or correlator receiver. (b) Simulate BPSK BER vs. SNR in Python/MATLAB; train a simple ML regression model to predict BER from SNR values and compare with the theoretical BER curve.

14. **Exp 6 (BFSK):** (a) Use AI/LLM tools to explain BFSK modulation/demodulation, the orthogonality condition between the two frequencies, and compare coherent vs. non-coherent BFSK detection. (b) Train a binary classifier to identify the transmitted frequency (f_1 or f_2) from BFSK signal features; simulate BER performance and compare with theoretical values.
15. **Exp 7 (QPSK):** (a) Use AI/LLM tools to explain the QPSK constellation diagram, the mapping of 2-bit symbols to phase values, the spectral efficiency advantage over BPSK, and coherent demodulation using a quadrature receiver. (b) Simulate QPSK in Python/MATLAB and plot the constellation diagram; train a multi-class classifier (KNN or SVM) to classify received symbols from I-Q data points.
16. **Exp 8 (DPSK):** (a) Use AI/LLM tools to explain differential encoding in DPSK, why DPSK does not require a coherent carrier reference, and compare its BER performance with coherent BPSK. (b) Train a binary classification model to detect phase changes (+180° vs. 0°) from DPSK signal features, simulating the differential decoding operation.

AI Integration	AI integration exercises span both sections. Section A tasks apply regression and classification to analog waveform parameters (modulation index, pulse width, pulse position, receiver metrics); Section B tasks apply binary and multi-class classifiers and regression models to predict BER, detect aliasing, estimate SQNR, and classify QPSK constellation symbols. All Python exercises are to be submitted as Google Colab .ipynb notebooks. LLM-assisted derivations (theory tasks) are to be documented in the lab record with source prompts and AI-generated explanations annotated by the student.

Recommended Free & Open-Source AI / Computational Tools
Python (NumPy, Matplotlib, SciPy, scikit-learn) on Google Colab (free) – signal simulation, BER analysis, ML classifiers, regression models
MATLAB / GNU Octave (free) – analog and digital modulation simulation, constellation diagrams, BER vs. SNR plots
scikit-learn (free, open-source) – Logistic Regression, KNN, SVM, Random Forest for BER prediction and symbol classification
Virtual Labs – VLSI & Communication Labs, IIT / IIIT (free): https://vlabs.ac.in – simulation of AM, FM, PCM, and digital modulation experiments
LibreOffice Calc (free, open-source) – standardised data recording templates for all experiments

Instructions

1. Faculty members handling the laboratory are requested to instruct students to design and construct circuits manually and not to use readymade kits.
2. The suggested AI/ML tasks may be modified based on the decision of the subject expert and laboratory availability.
3. Students must submit the lab record for every experiment at the next session; AI exercise notebooks (.ipynb) to be submitted via Google Classroom or institutional LMS.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Implement and verify analog modulation and demodulation circuits for AM, DSB-SC, FM, PAM, PWM, PPM, and FDM; measure modulation index, frequency deviation, channel bandwidth, and evaluate sensitivity and selectivity of a radio receiver.	L3 (Apply)
CO2	Verify the Nyquist sampling theorem and implement TDM; implement Delta Modulation and PCM systems; observe granular noise, slope overload, and quantisation error; measure SQNR for varying bit depths.	L3–L4 (Apply, Analyse)
CO3	Implement BPSK, BFSK, QPSK, and DPSK modulation and demodulation; plot BER vs. SNR curves; draw and interpret constellation diagrams; verify differential encoding and coherent/non-coherent detection.	L4 (Analyse)
CO4	Evaluate and compare analog and digital modulation schemes in terms of bandwidth efficiency, noise immunity, BER performance, and power requirements; assess practical limitations in circuit implementations.	L4–L5 (Analyse, Evaluate)
CO5	Apply AI/LLM tools to derive modulation expressions and explain communication system theory; develop Python/MATLAB-based ML regression and classification models to predict BER, estimate modulation parameters, detect aliasing, and classify modulation symbols.	L5–L6 (Evaluate, Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state laws	CO1–CO4	State Carson's rule; define modulation index; state Nyquist criterion; define BER; recall SQNR formula; define sensitivity and selectivity
L2 – Understand	Explain systems, describe procedures	CO1–CO5	Explain envelope detector operation; describe coherent vs. non-coherent detection; explain differential encoding in DPSK; describe FDM guard bands

L3 – Apply	Construct circuits, record, compute	CO1, CO2	Implement AM circuit and measure modulation index; record BER for BPSK; perform PCM sampling and encoding; measure frequency deviation in FM
L4 – Analyse	Compare, verify, interpret results	CO2–CO4	Compare BER curves for BPSK, BFSK, QPSK; verify Nyquist theorem experimentally; analyse effect of step size on Delta Modulation distortion; compare TDM vs. FDM efficiency
L5 – Evaluate	Assess accuracy, evaluate models	CO4, CO5	Evaluate ML model accuracy for BER prediction; assess SQNR from regression model vs. theoretical expression; evaluate detection performance of QPSK classifier
L6 – Create	Automate, develop, propose	CO5	Code Python BER simulator for BPSK; develop QPSK symbol classifier using SVM; build aliasing detector using Logistic Regression; write regression script for SQNR prediction

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	2	2	3	2	1	1	1	2	2	–3	1	3	2
CO2	L3-L4	3	3	2	3	2	1	–3	1	2	2	–3	2	3	2
CO3	L4	3	3	3	3	2	1	–3	1	2	2	–3	2	3	2
CO4	L4-L5	3	3	3	3	3	1	1	1	2	2	–3	2	3	3
CO5	L5-L6	2	3	3	3	3	1	1	1	2	3	1	3	2	3

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
AI / ML Lab Exercises	6 exercises	10	CO5
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every experiment at the next session. AI exercises to be submitted as Google Colab .ipynb notebooks.

PCB DESIGN AND PROTOTYPE DEVELOPMENT

Course Code	Year / Sem	L	T	Pr	P	Credits	Category
26AEC04403S	II Year II Sem	1	0	0	2	2	Skill Enhancement Course – II

Prerequisites: Fundamentals of Basic Electronics; Component identification and symbols

Course Objectives

- Introduce PCB terminology, component symbols and footprints, and EDA tools for schematic creation, annotation, and netlist generation.
- Develop understanding of the PCB design flow — component placement, routing, DRC, track width selection, and artwork generation for single-side, double-side, and multilayer boards.
- Familiarise students with design-for-manufacturability (DFM) principles including layout design standards, general design factors for analog and digital circuits, and design-specification standards.
- Enable students to design open-ended PCB development boards for analog, digital, and mixed-signal applications, applying complete design-to-layout workflows.
- Develop competency in physical PCB fabrication — single-sided PCB etching, component mounting, and cabinet assembly — executing the complete cycle from schematic to finished prototype.

Theory Content

Unit I – Fundamentals of Basic Electronics and PCB Basics

1. Component identification: resistors, capacitors, inductors, diodes, transistors, ICs; recognition of component symbols and their PCB footprints.
2. Understanding and reading schematics; navigating EDA schematic editors.
3. Creating a new PCB project; browsing and selecting footprint libraries.
4. Setting up PCB layers; understanding layer stack-up for single, double, and multilayer boards.
5. Design Rule Checking (DRC): purpose, setup, and interpreting DRC violations.
6. Track width selection: current-carrying capacity, IPC-2221 guidelines, and impedance-controlled routing basics.
7. Component selection criteria: package types, availability, thermal and electrical ratings.
8. Routing fundamentals: manual and auto-routing, clearance rules, via placement, and design completion.

Unit II – Introduction to PCB

9. Definition and need/relevance of Printed Circuit Boards in modern electronics.

10. Background and history of PCB: evolution from point-to-point wiring to advanced HDI boards.
11. Types of PCB: single-sided, double-sided, multilayer, rigid, flexible, and rigid-flex.
12. Classes of PCB design: Class 1 (General Electronics), Class 2 (Dedicated Service), Class 3 (High Reliability).
13. Terminology in PCB design: trace, pad, via, annular ring, silk screen, solder mask, gerber, drill file, BOM.
14. Different Electronic Design Automation (EDA) tools: KiCad, Altium Designer, Eagle, OrCAD, Proteus — features and comparison.

Unit III – PCB Design Process

15. PCB design flow: concept → schematic → netlist → placement → routing → DRC → gerber generation → fabrication.
16. Placement and routing strategies: component clustering, decoupling capacitor placement, power and ground plane routing.
17. Steps involved in layout design: board outline, layer assignment, component placement, trace routing, copper pour.
18. Artwork generation methods: manual tape-up (legacy) and CAD-based gerber/drill file generation.
19. General design factors for digital circuits: signal integrity, crosstalk, return current paths, decoupling, clock routing.
20. General design factors for analog circuits: noise minimization, ground plane separation, component orientation, shielding.
21. Layout and artwork for single-side, double-side, and multilayer boards.
22. Design for manufacturability (DFM): panelisation, fiducial markers, test points, solder mask expansion, minimum feature sizes.
23. Design-specification standards: IPC-2221, IPC-7351 (footprint standards), IPC-A-610 (acceptability of assemblies).

Practice Exercises (Any Twelve to be Performed)

Students are required to perform any twelve of the following exercises using an approved EDA tool (KiCad / Eagle / Altium / Proteus). Exercises include schematic creation, PCB layout, and physical fabrication.

1. **PCB Design Tool Familiarisation (Mandatory)**
   - Schematic Design: familiarisation of the Schematic Editor; schematic creation, annotation, and netlist generation.
   - Layout Design: familiarisation of the Footprint Editor; mapping of components; creation of PCB layout from schematic.
   - Create new schematic components (custom symbols).
   - Create new component footprints (custom land patterns).
2. Voltage Regulator circuit using IC 7805: schematic, PCB layout with proper bypass capacitors, DRC, and gerber generation.

3. Inverting Amplifier or Summing Amplifier using Op-Amp (e.g., LM741 / TL071): schematic, PCB layout with power supply decoupling.
4. Full-Wave Rectifier with filter capacitor: schematic, PCB layout, component footprint mapping, DRC.
5. Astable Multivibrator using IC 555: schematic, PCB layout, frequency calculation, and layout verification.
6. Monostable Multivibrator using IC 555: schematic, PCB layout, pulse-width calculation, and layout verification.
7. RC Phase-Shift Oscillator or Wein-Bridge Oscillator using Transistor: schematic, PCB layout, frequency design.
8. Full-Adder using Half-Adders: logic schematic, PCB layout for the combinational circuit, DRC and routing.
9. 4-Bit Binary / MOD-N Counter using D Flip-Flops: schematic, PCB layout, clock routing, and DRC.
10. Development Board for an open-ended Analog Application: student-defined analog circuit; full design flow from schematic to routed PCB.
11. Development Board for an open-ended Digital Application: student-defined digital circuit; full design flow from schematic to routed PCB.
12. Open-Ended Experiment (Analog / Digital / Mixed-Signal): a circuit of similar nature and magnitude assigned by the teacher; student to independently design, lay out, and simulate / execute.
13. Power Section Development Board: design a PCB incorporating IC 7805, filter capacitors, current-limiting resistor, power-on LED indicator, and input/output headers; generate gerber files.
14. **PCB Fabrication and Assembly (Mandatory Capstone Exercise):** fabricate a single-sided PCB for any one of the above circuits; mount components and assemble the complete board in a cabinet. Document the full process: artwork printout, UV exposure / chemical etching, drilling, soldering, and functional testing.

Course Outcomes (COs)

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Use EDA tools to create schematics, annotate designs, generate netlists, map component footprints, set up PCB layers and design rules, and complete routed PCB layouts for standard analog and digital circuits.	L3 (Apply)
CO2	Design manufacturable single-side and double-side PCB layouts for regulators, amplifiers, oscillators, multivibrators, and counters, applying track width selection, DRC, routing best practices, and artwork/gerbergeneration.	L3–L4 (Apply, Analyse)
CO3	Apply design-for-manufacturability principles and design-specification standards (IPC-2221, IPC-7351) to evaluate and optimise PCB layouts for	L4–L5 (Analyse,

	analog and digital circuits, identifying and resolving design rule violations.	Evaluate)
CO4	Independently design open-ended PCB development boards for analog, digital, and mixed-signal applications, executing the full workflow from problem definition through schematic, footprint mapping, placement, routing, and gerber generation.	L5–L6 (Evaluate, Create)
CO5	Fabricate and assemble a single-sided PCB prototype — including artwork generation, chemical etching, drilling, component mounting, and soldering — and verify functional operation of the assembled circuit in a cabinet.	L6 (Create)

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Viva Questions
L1 – Remember	Recall definitions, state standards	CO1–CO4	Define via, pad, annular ring, solder mask; state IPC-2221 class definitions; name EDA tools; list PCB types; recall track width formula
L2 – Understand	Explain procedures, describe flow	CO1–CO5	Explain PCB design flow from schematic to gerber; describe DFM principles; explain why decoupling capacitors are placed close to ICs; describe DRC purpose
L3 – Apply	Create schematics, route PCBs	CO1, CO2	Create 7805 regulator schematic and layout; route a full-adder PCB; generate gerber files for a rectifier circuit; map custom footprint to component
L4 – Analyse	Evaluate layouts, resolve DRC errors	CO2, CO3	Identify and fix DRC clearance violations; compare auto-routed vs manually routed trace quality; analyse ground plane continuity; evaluate track width for given current
L5 – Evaluate	Assess DFM compliance, optimise	CO3, CO4	Evaluate a PCB layout against IPC-2221 requirements; assess manufacturability of an open-ended design; compare layout options for noise minimisation in analog circuits
L6 – Create	Fabricate, assemble, prototype	CO4, CO5	Design and fabricate a complete single-sided PCB prototype; independently execute open-ended development board design; assemble and test a functional circuit in a cabinet

CO – PO / PSO Articulation Matrix (NBA)

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

CO	Bloom's	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L3	3	2	2	2	3	1	–	1	2	2	–	2	3	2
CO2	L3-L4	3	3	3	2	3	1	–	1	2	2	–	2	3	3
CO3	L4-L5	3	3	3	3	3	1	1	1	2	2	1	2	3	3
CO4	L5-L6	2	3	3	3	3	2	1	1	3	3	2	3	3	3
CO5	L6	2	2	3	3	2	2	1	1	3	3	2	3	2	3

Recommended EDA and Fabrication Tools

Recommended Free & Open-Source EDA / Fabrication Tools
KiCad EDA (free, open-source) – schematic editor, PCB layout, 3D viewer, gerber generation; recommended primary EDA tool
Eagle CAD (free for students) – schematic and layout editor; widely used in industry; extensive component libraries
Proteus Design Suite (licensed / institutional) – schematic, PCB layout, and circuit simulation in one environment
FreePCB / pcb-rnd (free, open-source) – lightweight PCB editors for simple single-layer and double-layer boards
PCBWay / JLCPCB online Gerber viewers (free) – verify fabrication files before submission; check layer stack-up online
LibreOffice Calc (free) – Bill of Materials (BOM) preparation and component tracking

Instructions

1. Students are expected to design and construct circuits manually using EDA tools and not to use readymade templates or pre-routed layouts for graded exercises.
2. Exercise 1 (Tool Familiarisation) and Exercise 14 (Fabrication) are mandatory; any ten additional exercises are to be completed from Exercises 2–13.
3. The open-ended experiment (Exercise 12) is to be assigned by the teacher at the beginning of the semester; the student is expected to solve and execute independently without assistance.
4. All PCB layouts must pass DRC with zero errors before submission of the lab record.
5. Gerber files and drill files must be generated and submitted along with the layout file for each experiment.
6. For the fabrication exercise (Exercise 14), students must document the complete process with photographs at each stage: artwork, etching, drilling, soldering, and final assembly.

Assessment Pattern (R26)

Assessment Component	Frequency / Duration	Marks	COs Assessed
Lab Internal (Record + Viva)	Continuous	25	CO1–CO4
Mid-Term Practical Test	1 test	15	CO1, CO2, CO3
Open-Ended Design Exercise	1 exercise (Expt. 10/11/12)	10	CO4
End Semester Practical Exam	3 hrs	35	CO1–CO4
End Semester Viva Voce	30 min	15	All COs
TOTAL		**100**	

Note: Lab record to be submitted for every exercise at the next session. Fabrication exercise record to include photographic documentation and functional test results.

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

Core Content:

Multidisciplinary Nature of Environmental Studies: definition, scope, and importance — need for public awareness. Natural Resources: renewable and non-renewable resources — forest resources (use, over-exploitation, deforestation, case studies), water resources (use and over-utilisation, floods, drought, conflicts over water, dams), mineral resources (environmental effects of extracting and using mineral resources), food resources (world food problems, effects of modern agriculture, fertilizer-pesticide problems), energy resources (use and exploitation of conventional and non-conventional energy sources).

AI Integration:

Students use Google Earth Engine for deforestation and land-use change mapping, Python (scikit-learn) for groundwater level prediction, and ML forecasting tools for renewable energy output analysis.

Free/Open-Source AI Tools:

Google Earth Engine (free cloud GIS); QGIS (free, open-source); Python (scikit-learn, Matplotlib) on Google Colab (free); Global Forest Watch (free dashboard); AI tools: ChatGPT / Gemini

UNIT II	Ecosystems and Biodiversity and its Conservation

Core Content:

Ecosystems: concept, structure and function — producers, consumers and decomposers — energy flow, ecological succession, food chains, food webs and ecological pyramids. Types, characteristic features, structure and function of: forest, grassland, desert, and aquatic ecosystems. Biodiversity and its Conservation: definition, bio-geographical classification of India, value of biodiversity, India as a mega-diversity nation, hot-spots, threats to biodiversity, endangered and endemic species of India, in-situ and ex-situ conservation.

AI Integration:

Students use iNaturalist for AI-assisted species identification during a campus biodiversity survey, MaxEnt for biodiversity hotspot mapping, and Google Earth Engine for NDVI-based ecosystem health comparison.

Free/Open-Source AI Tools:

iNaturalist (free app/web); MaxEnt (free species distribution model); Google Earth Engine (free); QGIS (free); Python on Google Colab (free)

UNIT III Environmental Pollution and Solid Waste Management

Core Content:

Environmental Pollution: definition, cause, effects and control measures of air pollution, water pollution, soil pollution, marine pollution, noise pollution, thermal pollution, and 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:

Students build an ML-based Air Quality Index prediction model using Python (scikit-learn, TensorFlow), analyse IoT water quality sensor data for anomaly detection, and use QGIS/Python for disaster risk zone mapping.

Free/Open-Source AI Tools:

OpenAQ (free global AQ data API); QGIS (free); Python (scikit-learn, TensorFlow, Matplotlib) on Google Colab (free); CPCB data portal (free); AI tools: ChatGPT / Gemini

UNIT IV Social Issues and the Environment

Core Content:

From unsustainable to sustainable development. Urban problems related to energy. Water conservation, rainwater harvesting, watershed management. Resettlement and rehabilitation of people — problems and case studies. Environmental ethics: issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion — case studies. Wasteland reclamation. Consumerism and waste products. Environment Protection Act, Air and Water Pollution Control Acts, Wildlife Protection Act, Forest Conservation Act — issues in enforcement. Public awareness.

AI Integration:

Students analyse CMIP6 climate model data with Python (xarray, Matplotlib), estimate carbon footprints using scikit-learn, and use QGIS/Python for watershed delineation and rainwater harvesting potential mapping.

Free/Open-Source AI Tools:

Python (xarray, Matplotlib, scikit-learn) on Google Colab (free); CMIP6 climate data (free, ESGF portal); QGIS (free); AI tools: ChatGPT / Gemini

UNIT V Human Population and the Environment

Core Content:

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); visit to a local polluted site; study of common plants, insects, and birds.

AI Integration:

Students use Python regression and LSTM models for population growth forecasting, explore AI in public health surveillance, and use iNaturalist/eBird with geo-tagging during field work to document environmental assets.

Free/Open-Source AI Tools:

iNaturalist (free); eBird (free); QGIS (free); Python (scikit-learn, TensorFlow) on Google Colab (free); OpenStreetMap (free); AI tools: ChatGPT / Gemini

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Recall and explain the multidisciplinary nature of environmental studies and describe the uses, over-exploitation, and environmental problems of forest, water, mineral, food, and energy resources.	L1 (Remember) L2 (Understand)
CO2	Describe the structure and function of different ecosystem types and outline the concept, value, threats, and conservation strategies of biodiversity.	L2 (Understand) L3 (Apply)
CO3	Identify causes, effects, and control measures of air, water, soil, marine, noise, thermal, and nuclear pollution; apply solid waste management principles; analyse pollution case studies and disaster management strategies.	L3 (Apply) L4 (Analyze)
CO4	Analyse social issues related to unsustainable development; evaluate impacts of climate change and ozone depletion; examine Indian environmental legislation and assess enforcement challenges.	L4 (Analyze) L5 (Evaluate)
CO5	Describe population growth trends and environmental impacts; demonstrate the role of IT and AI tools in environmental monitoring and field data collection.	L2 (Understand) L3 (Apply)

CO – PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	L1,L2	2	1	–	–	1	3	3	2	–	2	–	2	2	1
CO2	L2,L3	2	2	–	1	2	3	3	2	1	2	–	2	2	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

Textbooks

1. Bharucha E., Textbook of Environmental Studies for Undergraduate Courses, Universities Press.
2. Palaniswamy, Environmental Studies, Pearson Education.

Reference Books

1. Dave D. and Reddy E.S.B., Textbook of Environmental Science, Cengage Publications.
2. Masters G.M. and Ela W.P., Introduction to Environmental Engineering and Science, Prentice Hall of India.

Web Resources & NPTEL Links

- NPTEL: Environmental Science – https://nptel.ac.in/courses/124107002
- Google Earth Engine – https://earthengine.google.com
- Global Forest Watch – https://www.globalforestwatch.org
- OpenAQ – https://openaq.org
- iNaturalist – https://www.inaturalist.org
- QGIS – https://qgis.org