ADITYA COLLEGE OF ENGINEERING

(An Autonomous Institution)

www.acem.ac.in

DEPARTMENT OF CIVIL ENGINEERING

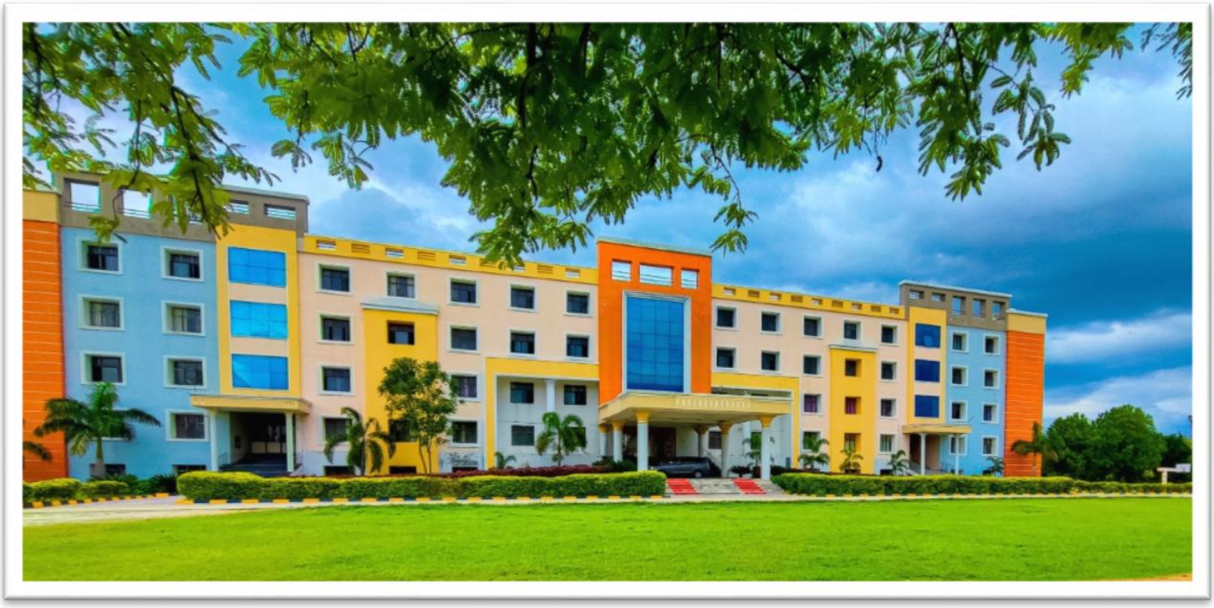
Course Structure
&
Detailed Syllabus

For the students admitted to

B. Tech. Regular Four Year Degree Programme from the Academic Year 2026-27

and

B. Tech. Lateral Entry Scheme from the Academic Year 2027-28



ADITYA COLLEGE OF ENGINEERING, MADANAPALLE

(UGC - Autonomous Institution & Accredited with NAAC A+ Grade)

Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu

Madanapalle - 517325, Annamayya (Dist.), A. P.

www.acem.ac.in

ACEM - R26 REGULATION

CIVIL ENGINEERING

COURSE STRUCTURE AND SYLLABUS

for

B. Tech. (Regular – Full Time)

(Effective for the Students Admitted into I Year from the Academic Year 2026 - 27 onwards)

and

B. Tech. (Lateral Entry Scheme)

(Effective for the Students getting Admitted into II Year through Later Entry Scheme from the Academic Year 2027 - 28 onwards)

Institute Vision:

To impart quality in engineering education to meet the technological advances and industrial requirements with global standards.

Institute Mission:

- ❖ Provide quality technical education through skill-based trainings and promote research and development, and consultancy services.
- ❖ Offer state-of-the-art-infrastructure for supporting technological advances.
- ❖ Develop disciplined, creative and globally competent engineers.
- ❖ Equip and empower the faculty at all levels to promote innovations and technical advancements in various domains of engineering.

Department Vision:

To be a globally recognized civil engineering department that provides excellent education and research for building innovative infrastructure and sustainable communities.

Department Mission:

- ❖ Fostering excellence in teaching and learning through a balanced curriculum of theoretical science, technical skills, and practical application.
- ❖ Promoting innovative thinking and the use of emerging technologies for sustainable infrastructure design and construction.
- ❖ Instilling in students a strong foundation in engineering principles, professional ethics, and civic responsibility to make positive contributions to society as civil engineers.

Programme Outcomes (POs):

On Successful completion, the graduate will be able to,

- ❖ **PO1_Engineering Knowledge:** Apply knowledge of mathematics, natural sciences, engineering fundamentals, and an engineering specialization to a wide range of practical procedures and practices.
- ❖ **PO2_Problem Analysis:** Identify and analyse well-defined engineering problems and reach substantiated conclusions using codified methods of analysis specific to their field of activity.
- ❖ **PO3_Design/Development of Solutions:** Design solutions for well-defined technical problems and assist in the design of systems, components, or processes to meet specified needs, with appropriate consideration for public health and safety, as well as cultural, societal, and environmental factors.
- ❖ **PO4_Conduct Investigations of Complex Problems:** Conduct investigations of well-defined problems by locating and searching relevant codes and catalogues, and by conducting standard tests and measurements.
- ❖ **PO5_Engineering Tool Usage:** Apply appropriate techniques, resources, and modern computing, engineering and IT tools to well-defined engineering problems, while being aware of their limitations.

- **PO6_The Engineer and the World:** When solving well-defined engineering problems, evaluate sustainable development impacts to society, the economy, sustainability, health and safety, legal frameworks, and the environment.
- **PO7_Ethics:** Understand and commit to professional ethics and the norms of technician practice, including compliance with relevant laws. Demonstrate an understanding of the need for diversity and inclusion.
- **PO8_Individual and Collaborative Team Work:** Function effectively as an individual, and as a member or leader in diverse and inclusive teams, in multidisciplinary face-to-face/remote/distributed settings.
- **PO9_Communication:** Communicate effectively and inclusively on well-defined engineering activities with the engineering community and society at large by comprehending the work of others, documenting one's own work, and giving and receiving clear instructions.
- **PO10_Project Management and Finance:** Demonstrate awareness of engineering management principles as a member or leader of a technical team and apply them to manage projects in multidisciplinary environments.
- **PO11_Life-Long Learning:** Recognize the need for, and have the ability for, independent updating in the face of specialized technical knowledge.

Program Specific Outcomes (PSOs):

On Successful completion, the graduate (Civil Engineering) will be able to,

- **PSO1:** Analyze and design civil engineering structures using knowledge of structural analysis, mechanics, materials, and safety standards. Design and evaluate structural systems like buildings, bridges, dams, and transportation infrastructure.
- **PSO2:** Apply engineering principles to plan, design, construct, and maintain civil engineering works, including surveying, construction planning, project management, and infrastructure maintenance.

Programme Educational Objectives (PEOs):

After few years of graduation, the graduates of Civil Engineering are expected to

- **PEO1:** Apply knowledge of mathematics, science, and engineering principles to identify, formulate, and solve civil engineering problems.
- **PEO2:** Students will gain strong analytical and problem - solving skills to design civil engineering systems like bridges, roads, dams, and buildings.
- **PEO3:** Students will become proficient in using software like Auto CAD, STAAD, and MATLAB to plan, execute, and interpret experiments and analyse data.
- **PEO4:** Work professionally with effective communication, teamwork, and leadership skills. Students will practice collaboration, written and oral communication, project management, and ethical decision-making.

B. Tech. – Civil Engineering
Course Structure & Syllabus – ACEM R26 Regulations
(Applicable from the Academic Year 2026-27 onwards)

INDUCTION PROGRAMME

S.No.	Name of the Course	Category	L-T-P-C
1.	Physical Activities -- Sports, Yoga and Meditation, Plantation	MC	0-0-6-0
2.	Career Counselling	MC	2-0-2-0
3.	Orientation to all branches -- career options, tools, etc.	MC	3-0-0-0
4.	Orientation on admitted Branch -- corresponding labs, tools and platforms	EC	2-0-3-0
5.	Proficiency Modules & Productivity Tools	ES	2-1-2-0
6.	Assessment on basic aptitude and mathematical skills	MC	2-0-3-0
7.	Remedial Training in Foundation Courses	MC	2-1-2-0
8.	Human Values & Professional Ethics	MC	3-0-0-0
9.	Communication Skills -- focus on Listening, Speaking, Reading, Writing skills	BS	2-1-2-0
10.	Concepts of Programming	ES	2-0-2-0

Group - A

B. Tech. – I Year I Semester

S. No.	Category	Course Code	Name of the Course	L	T	Pr.	P	Credits
1.	BS	26BSHB101T	Linear Algebra & Calculus	3	0	2	0	4
2.	BS	26BSHB103T	Physics for Civil Engineers	2	1	0	0	3
3.	ES	26ES03101T	Engineering Graphics	1	0	4	0	3
4.	ES	26ES05101T	Computational Thinking & Problem Solving with Python	3	0	0	0	3
5.	ES	26ES31101T	AI Tools and Applications	1	0	4	0	3
6.	BS	26BSHB103P	Physics for Civil Engineers Lab	0	0	0	3	1.5
7.	ES	26ES05101P	Computational Thinking & Problem Solving with Python Lab	0	0	0	3	1.5
8.	ES	26ES03102P	Engineering Workshop	0	0	0	3	1.5
9.	HM	26HMHS103L	NSS / NCC / Scouts & Guides / Community Service	0	0	0	1	0.5
			Total	**10**	**1**	**10**	**10**	**21**

B. Tech. – I Year II Semester

S. No.	Category	Course Code	Name of the Course	L	T	Pr.	P	Credits
1.	BS	26BSHB102T	Differential Equations & Vector Calculus	3	0	2	0	4
2.	BS	26BSHB107T	Chemistry for Civil Engineers	3	0	0	0	3
3.	HM	26HMHS101T	Communicative English	2	0	0	0	2
4.	ES	26ES03103T	Design Thinking	1	0	4	0	3
5.	PC	26PC01101T	Engineering Mechanics	2	0	2	0	3
6.	BS	26BSHB107P	Chemistry for Civil Engineers Lab	0	0	0	3	1.5
7.	HM	26HMHS101P	Communicative English Lab	0	0	0	2	1
8.	PC	26PC01102P	Civil Engineering Workshop	0	0	0	2	1
9.	HM	24HMHS102L	Health and Wellness, Yoga and Sports	0	0	0	1	0.5
10.	MC	26ES05102A	Cybersecurity Awareness for Engineers	2	0	0	0	0
			Total	**11+2**	**0**	**8**	**8**	**19**

B. Tech. – II Year I Semester

S. No.	Category	Course Code	Name of the Course	L	T	Pr.	P	Credits
1.	BS	26BSHB211T	Complex Variables and Probability Theory	2	1	0	0	3
2.	HM	26HMHS205T	Managerial Economics and Financial Analysis	2	0	0	0	2
3.	ES	26ES01201T	Sustainability and Green Engineering (AI Integrated)	2	0	0	0	2
4.	EC	26EC02201T	Basic Electrical & Electronics Engineering	3	0	0	0	3
5.	PC	26PC01203T	Surveying (AI Integrated)	2	0	2	0	3
6.	PC	26PC01204T	Strength of Materials (AI Integrated)	2	0	2	0	3
7.	EC	26EC02201P	Basic Electrical & Electronics Engineering Lab	0	0	0	2	1
8.	PC	26PC01203P	Surveying Lab (AI Integrated)	0	0	0	3	1.5
9.	PC	26PC01204P	Strength of Materials Lab	0	0	0	3	1.5
10.	SE	26SE01201S	Building Planning and Drawing (AI Integrated)	1	0	0	2	2
11.	MC	26BSHB214A	Environmental Science	2	0	0	0	0
			Total	**14+2**	**1**	**4**	**10**	**22**

B. Tech. – II Year II Semester

S. No.	Category	Course Code	Name of the Course	L	T	Pr.	P	Credits
1.	BS	26BSHB213T	Biology for Engineers	2	0	0	0	2
2.	HM	26HMHS204T	Universal Human Values – Understanding Harmony & Ethical Human Conduct	3	0	0	0	3
3.	PC	26PC01205T	Fluid Mechanics (AI Integrated)	2	0	2	0	3
4.	PC	26PC01206T	Concrete Technology (AI Integrated)	2	0	2	0	3
5.	PC	26PC01207T	Structural Analysis (AI Integrated)	2	0	2	0	3
6.	PC	26PC01208T	Environmental Engineering (AI Integrated)	2	0	2	0	3
7.	PC	26PC01205P	Fluid Mechanics Lab	0	0	0	3	1.5
8.	PC	26PC01206P	Concrete Technology Lab	0	0	0	3	1.5
9.	PC	26SE01202S	Auto CAD for Civil Engineers	1	0	2	0	2
10.	MC	26HMHS206A	Technical Paper Writing and IPR	2	0	0	0	0
			Total	**14+2**	**0**	**10**	**6**	**22**
	Mandatory Community Service Project (26IPIN301L) of 08 Weeks duration during Summer Vacation							

B.Tech. – I Year
I Semester

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB101T	BS	**Linear Algebra & Calculus** (Common to all branches of Engineering)	3	0	2	0	4

Pre-Requisites: Basic algebra, trigonometry, coordinate geometry, differential and integral calculus, vectors, matrices, determinants, and functions of several variables.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply matrix algebra techniques to solve systems of linear equations and analyze matrix properties relevant to engineering models. (L3)

CO2: Analyze eigenvalue problems, diagonalization, and quadratic forms to interpret linear transformations and engineering systems. (L4)

CO3: Apply mean value theorems, Taylor and Maclaurin series, and curvature concepts to approximate and interpret engineering functions. (L3)

CO4: Analyze multivariable functions using partial derivatives, Jacobians, gradients, and optimization techniques. (L4)

CO5: Evaluate areas, volumes, and coordinate transformations using multiple integrals in engineering contexts. (L5)

CO – PO Articulation Matrix:

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

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

Unit I: Matrix Algebra and Linear Systems:

Core Content: Matrices, operations, transpose, inverse, types of matrices, symmetric and skew-symmetric matrices; orthogonal, Hermitian, skew-Hermitian and unitary 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;

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; Properties, 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; Nature of Quadratic form;

AI Integration: AI-assisted eigenvalue computation and verification; Python/SciPy visualization of eigen modes 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.

Text Books:

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

Reference Books:

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

Web Resources & NPTEL Links:

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

Recommended Free & Open-Source AI/Computational Tools:

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

B. Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB103T	BS	Physics for Civil Engineers	2	1	0	0	3

Pre-Requisites: Engineering Mathematics, Basics of Physics, Newton's laws, and Ray optics.

Course Objectives:

- To help students understand the concepts of interference, diffraction, and polarization and their applications in optical measurements, surveying, and civil engineering structures, with basic AI-assisted analysis techniques.
- To introduce crystallography and X-ray diffraction methods for studying crystal structures and construction materials with simple AI-assisted material characterization concepts.
- To provide knowledge on quantum mechanics and quantum computing concepts, along with basic AI-assisted quantum simulation and secure communication applications.
- To familiarize students with oscillations, vibration control, and acoustics of buildings used in earthquake- resistant and smart structures with simple AI-assisted monitoring techniques.
- To impart knowledge of ultrasonics and non-destructive testing methods for quality inspection and defect detection in civil engineering structures with basic AI-assisted inspection concepts.

Course Outcomes (COs):

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

CO1: Analyze the principles of interference, diffraction, and polarization and apply basic AI- assisted optical techniques for surveying, surface analysis, and civil engineering applications. (L4)

CO2: Examine crystal structures and interpret X-ray diffraction patterns for material Characterization using basic AI-assisted analysis techniques in civil engineering materials. (L4)

CO3: Apply the concepts of quantum mechanics and quantum computing to explain quantum-based communication systems with basic AI-assisted quantum simulations. (L3)

CO4: Analyze oscillatory systems, vibration control mechanisms, and architectural acoustics for civil engineering structures using basic AI-assisted monitoring and acoustic analysis methods. (L4)

CO5: Evaluate ultrasonic and non-destructive testing techniques for identifying defects and assessing the quality of structural materials using basic AI-assisted defect detection approaches. (L5)

CO – PO Articulation Matrix:

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

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

Unit I: Applied Wave Optics:

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 and Optical Surveying: 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: Measurement of distances, angles and surface alignment in civil engineering (qualitative).

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

AI Integration: AI-Assisted Flatness Detection Using Interference Fringes, AI-Based Mineral Analysis and Degradation Detection in Masonry and Concrete Using Diffraction.

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

Unit II: Crystallography and AI-Driven Material Characterization:

Core Content: Crystallography: Space lattice, Basis, Unit Cell and lattice parameters – Crystal systems, Bravais Lattices — Coordination number - Packing fraction of SC, BCC & FCC - Miller indices – Separation between successive (h k l) planes.

X-ray diffraction: Bragg's law - X-ray Diffractometer – Crystal structure determination by Laue's method. Application: Checking the quality of construction materials.

AI Integration: AI-driven XRD for Real-time corrosion assessment in reinforced concrete and identifying micro-cracks in ageing infrastructure.

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

Unit III: Introduction to Quantum Mechanics & Quantum Computing:

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

Introduction to Quantum Computing: Classical bits vs qubits. Concept of superposition and Entanglement, Basic Quantum Operations: Hadamard gate (idea of superposition), Pauli-X gate, Basic idea of quantum measurement, Applications: Basic idea of quantum cryptography and secure communication.

AI Integration: AI-assisted quantum circuit simulation using IBM Qiskit (free). Python-based visualization of wave functions and quantum probability densities.

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

Unit IV: Oscillations and Acoustics of Buildings:

Core Content: Introduction to oscillations, Oscillations of springs, Series and parallel combinations of springs (quantitative), Stiffness factor and its physical significance. Engineering applications of springs in Base Isolation Systems for earthquake-resistant buildings. Damping and Resonance: Theory of damped oscillations (Over, Under and Critical damping- all qualitative); Forced oscillations (qualitative) and Resonance. Sharpness of resonance and Quality factor (Q). Applications: Tuned Mass Dampers (TMDs) for seismic resilience and wind-induced vibration control in skyscrapers (qualitative).

Acoustics of Buildings: Introduction to architectural acoustics; Reverberation and reverberation time, Sabine's formula (qualitative). Factors affecting architectural acoustics and requirements for acoustically optimized halls. Applications- Active Noise Cancellation (ANC) systems for smart urban facades (Qualitative).

AI Integration: AI-Assisted Strategies for noise mitigation in Green Buildings. Use of Acoustic Simulation AI for auditorium design.

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

Unit V: Ultrasonics and Non-Destructive Testing:

Core Content: Introduction to ultrasonics: Production of ultrasonics: Piezoelectric method and Magnetostriction method. Detection of ultrasonics: Kundt's tube and Quartz crystal methods. Role of Ultrasonic Pulse Velocity (UPV**)** testing in assessing concrete quality, uniformity and depth of cracks in structural elements.

Non-Destructive Testing (NDT): Introduction, need and advantages in civil engineering structures. Basic inspection methods: Visual inspection and AI-based drone inspections for inaccessible structures (qualitative). Liquid penetrant testing: basic principle for detecting surface defects in steel. Application: Electromagnetic inspection: eddy current testing (to detect surface cracks, corrosion and defects in conductive materials). (basic idea)

AI Integration: AI-Powered Defect Detection in Structural Steel, Weld Joints and Corrosion in Reinforced Concrete (RCC)

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

Text Books:

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. Engineering Physics — B. K. Pandey & S. Chaturvedi — Cengage Learning, 2021
2. Engineering Physics — Shatendra Sharma &Jyotsna Sharma — Pearson, 2018
3. Engineering Physics — Sanjay D. Jain, D. Sahasrabudhe & Girish — Universities Press, 2010
4. Engineering Physics — M. R. Srinivasan — New Age International, 2009
5. Solid State Physics — S. O. Pillai — New Age International
6. Modern Physics — R. Murugeshan & Kiruthiga Sivaprasath — S. Chand

Web Resources & NPTEL Links:

1. NPTEL – nptel.ac.in
2. MIT Open Course Ware – ocw.mit.edu
3. Hyper Physics – hyperphysics.phy-astr.gsu.edu
4. Feynman Lectures – feynmanlectures.caltech.edu
5. PhET Simulations – phet.colorado.edu

Recommended Free & Open-Source AI/Computational Tools:

1. Python (NumPy, SciPy, OpenCV, Matplotlib) – FREE on Google Colab for optics and EM simulations
2. IBM Qiskit (free, open-source) – For quantum computing and quantum cryptography simulations
3. scikit-learn (free, open-source) – For semiconductor data analysis and signal classification
4. PhET Interactive Simulations (free) – For wave optics, semiconductors and quantum phenomena

B. Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES03101T	ES	**Engineering Graphics** (Common to all branches of Engineering)	1	0	4	0	3

Pre-Requisites: Basic Geometry, and Fundamental Python Programming

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Construct engineering curves and scales using traditional drafting geometry validated by automated algorithmic paths. (L3)

CO2: Solve multi-planar projection instances of linear points, segments, and standard regular planar silhouettes. (L3)

CO3: Render projections of standard 3D solids and identify variations using structural boundary algorithms. (L4)

CO4: Draft complex cross-sectional geometries and mapped surface extensions of sliced geometric forms. (L4)

CO5: Produce composite isometric layouts alongside generative text-to-CAD system alternatives for dynamic prototyping. (L5)

CO – PO Articulation Matrix:

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

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

Unit 1: Introduction to Engineering Graphics, Curves, and Scales:

Core Content:

Introduction: Principles of Engineering Graphics, signboards, 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.

Hands-On Experience: Manual draft generation of an ellipse and diagonal scale, paired with a software verification loop.

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

Unit II: Orthographic Projections of Points, Lines:

Core Content: Principles of orthographic projections, projection of points in all four quadrants. Projection of straight lines inclined to one or both reference planes, traces of lines, determination of true lengths and true inclinations.
Projections of Planes: Regular planes Perpendicular to both reference planes, parallel to one reference plane and inclined to the other reference plane; plane inclined to both the reference planes.
Hands-On Experience & Application:
Project: Construct a transformation script where a user inputting 2D projection endpoint automatically outputs the 3D line representation, true spatial length, and inclination angles without manual geometric tracking.
Application: Calibration setups for stereoscopic robotic arms tracking trajectories across multiple camera planes.
AI Integration Module: Machine learning pipelines for 3D coordinate estimation from 2D views; multi-view structural tracking; training neural models to predict true line length and orientation angles based solely on coordinate projections.

Unit III: Projection of Regular Solids:
Core Content:
Projection of Regular Solids: Types of solids: Polyhedra and Solids of revolution. Projections of solids in simple positions: Axis perpendicular to horizontal plane, Axis perpendicular to vertical plane and Axis parallel to both the reference planes, Projection of Solids with axis inclined to one reference plane and parallel to another plane.
Hands-On Experience & Application:
Project: Set up a classification notebook using scikit-learn or a simple convolutional model to automatically identify the type of solid (e.g., hexagonal prism vs. cone) present within an orthographic engineering layout image.
Application: Automated component sorting inside factory staging bays and conveyor streams.
AI Integration Module: Object recognition models (e.g., YOLO / Mobile Net) trained to identify standard solid geometries; automated multi view projection generation from 3D CAD meshes using programmatic render layers.

Unit IV: Sections of Solids, Development of Surfaces, and AI Generative Design:
Core Content:
Sections of Solids: Perpendicular and inclined section planes, Sectional views and True shape of section, Sections of solids in simple position only.
Development of Surfaces: Methods of Development: Parallel line development and radial line development. Development of a cube, prism, cylinder, pyramid and cone.
Hands-On Experience & Application:
Project: Use a nesting optimization script (e.g., using genetic algorithms or standard heuristics) to arrange several custom flat-surface solid developments onto a finite raw sheet grid, actively minimizing material margins and cut lines.
Application: Sheet metal stamping pattern automation in automotive and aerospace chassis manufacturing.
AI Integration Module: 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.

Unit V: Isometric Projections, Conversions, and 2D-to-3D Deep Reconstruction:
Core Content: Principles of isometric projection, isometric scale, isometric views/projections of planes, simple and compound solids. Conversion of isometric views to orthographic views and vice-versa.
Computer Graphics: Creating 2D&3D drawings of objects including PCB and Transformations using Auto CAD (Not for end examination).

Hands-On Experience & Application:

Project: Use a pretrained depth estimation or pixel-to-voxel neural model to parse a set of front, top, and side drawings, compiling them directly into an interactive 3D point cloud or boundary surface rendering.

Application: Instantly generating 3D printable mechanical files directly from archival 2D layout drafting logs.

AI Integration Module: 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.

Text Books:

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

Reference Books:

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

Video Links & Online Lectures:

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

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES05101T	ES	**Computational Thinking & Problem Solving with Python** (Common to all branches of Engineering)	3	0	0	0	3

Pre-Requisites: Basic knowledge of computer fundamentals, logical reasoning, problem-solving, and elementary mathematics.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply computational thinking concepts to design algorithms, flowcharts, and basic Python programs. (L3)

CO2: Develop programs using decision-making and looping constructs to solve logical problems. (L3)

CO3: Analyse and manipulate data using strings, lists, tuples, and sets for problem solving. (L4)

CO4: Design modular programs using functions, recursion, and dictionaries to improve code reusability. (L4)

CO5: Implement file handling, exception handling, and basic object-oriented programming concepts in Python applications. (L3)

CO – PO Articulation Matrix:

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

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

Unit I: Computational Thinking and Programming Basics:

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 Tools Integration: Use Flowgorithm to design and simulate algorithms (e.g., largest of 3 numbers); draw flowcharts using Lucidchart; use Algorithm Visualizer to trace algorithm execution; use ChatGPT to convert pseudocode to Python and verify output.

Unit II: Decision Making and Looping:

Decision Control Statements: Boolean expressions; if, if-else, if-elif-else, nested if; conditional expressions (ternary operator).

Looping Statements: while loop, for loop, iteration techniques, nested loops, infinite loops; loop control — break, continue, pass; else with loops.

Practical Problem Solving: Prime number check, pattern programs using nested loops, menu-driven programs using control statements.

AI Tools Integration: Use Python Tutor to visualize if-else and loop execution step by step; use Thonny to debug loop-based programs; use Replit for online coding practice; use ChatGPT to generate test cases for control flow programs.

Unit III: Strings and Data Structures:
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 Tools Integration: Use NLTK and TextBlob to perform text analysis (sentiment, word frequency) on strings; use Pandas and NumPy for list and array operations; use ChatGPT to explain string slicing and list comprehensions with concrete examples.

Unit IV: Functions and Problem Solving:
Dictionaries: Creation, operations, methods; dictionary-based applications.
Functions: Built-in and user-defined functions; function definition and calling; arguments — positional, keyword, default, variable-length; scope of variables (local and global).
Recursion: Recursive functions — factorial, Fibonacci; lambda functions (anonymous functions); applications of functions in problem solving.
AI Tools Integration: Use Jupyter Notebook for interactive function experiments; use SymPy for symbolic mathematical expressions; use PyCharm for modular program development; use ChatGPT to trace recursive function calls step by step.

Unit V: File Handling, Exception Handling, and OOP:
Modules and Packages: Creating and importing modules; standard library modules.
File Handling: Opening, reading, writing, and closing files; file modes; working with text files; processing CSV and Excel files.
Exception Handling: Types of errors (syntax, runtime, logical); try, except, finally blocks; raising exceptions.
Introduction to OOP: Classes, objects, attributes, methods, constructors, self-keyword; basic applications of OOP in Python.
AI Tools Integration: Use Pandas and OpenPyXL to read and write CSV/Excel files; use Pytest for testing exception handling; use ChatGPT to explain OOP class design with real-world examples; use Django for a simple mini web application as a capstone project.

Text Books:

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

Reference Books:

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

Online Tools Used in this Course:

1. Algorithm & Flowchart: Flowgorithm – flowgorithm.org |Lucidchart – lucidchart.com | Algorithm Visualizer – algorithm-visualizer.org
2. Programming & Debugging: Python Tutor – pythontutor.com | Thonny – thonny.org |Replit – replit.com

3. Data & Scientific: Pandas – pandas.pydata.org | NumPy – numpy.org |SymPy – sympy.org
4. NLP & Text: NLTK – nltk.org |TextBlob – textblob.readthedocs.io
5. Development: Jupyter Notebook – jupyter.org | PyCharm – jetbrains.com/pycharm |OpenPyXL – openpyxl.readthedocs.io
6. NPTEL – Problem Solving through Programming in C / Python – nptel.ac.in

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES31101T	ES	**AI Tools and Applications** (Common to all branches of Engineering)	1	0	4	0	3

Pre-Requisites: Basic computer literacy, internet usage, information-search skills, and fundamental communication and problem-solving skills.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the fundamental concepts of Artificial Intelligence (AI) to explain its working principles and demonstrate the use of AI tools across text, image, audio, video, and code applications. (L3)

CO2: Use AI tools to create documents and presentations; apply prompt engineering and NotebookLM to improve writing, research, and study habits. (L4)

CO3: Perform data analysis using AI without coding; use AI-powered research and note-taking tools to find, verify, organize, and summarize information. (L4)

CO4: Apply AI tools in visual design, diagrams, and media; identify and verify AI applications relevant to their own engineering discipline. (L5)

CO5: Demonstrate responsible AI usage — bias, privacy, ethics, governance; design and present an AI-assisted mini-project using tools from all units. (L5)

CO – PO Articulation Matrix:

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

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

Unit I: Understanding AI and AI for Documents and Presentations:

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:

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:

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:

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:

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.

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:

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

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB103P	BS	Physics for Civil Engineers Lab	0	0	0	3	1.5

Pre-Requisites: Basic knowledge of mechanics, waves, optics, electricity, measurement techniques, and fundamental concepts of physics and mathematics.

Course Objectives:

- To provide students with practical knowledge of fundamental principles of optics, diffraction, interference, and spectroscopy through laboratory experiments.
- To develop experimental skills for measuring wavelength, refractive index, dispersive power, particle size, and optical parameters using appropriate instruments.
- To develop the ability to investigate and interpret mechanical oscillations, waves, stretched strings, springs, and torsional systems through experimental methods.
- To familiarize students with quantum and modern physics experiments, including the photoelectric effect, and enable them to determine and interpret fundamental physical constants.

Course Outcomes (COs):

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

CO1: Apply optical principles and experimental techniques to determine refractive index, wavelength, dispersive power, thickness, and radius of curvature of optical materials. (L3)

CO2: Apply concepts of wave optics, diffraction, interference, and acoustic waves to determine wavelengths, particle size, and ultrasonic velocity using appropriate experimental methods. (L3)

CO3: Analyze mechanical oscillations, stretched strings, coupled oscillators, torsional systems, and spring systems to determine relevant physical parameters and frequencies. (L4)

CO4: Evaluate experimental data using photoelectric and quantum principles to estimate fundamental constants such as Planck's constant and interpret experimental results. (L5)

CO – PO Articulation Matrix:

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

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

List of Experiments:

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

14. Determination of ultrasonic velocity in liquid (Acoustic grating)
15. Determination of Rigidity modulus of material of a wire-dynamic method (Torsional pendulum)

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

Web Resources:

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

Recommended AI Tools:

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

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES05101P	ES	**Computational Thinking & Problem Solving with Python Lab** (Common to all branches of Engineering)	0	0	0	3	1.5

Pre-Requisites: Basic computer literacy, logical reasoning, problem-solving skills, and elementary mathematics.

Course Objectives:

- To develop problem-solving skills through algorithms, flowcharts, and basic Python programming.
- To enable students to implement decision-making and iterative solutions using Python control structures.
- To develop programming skills for manipulating strings and built-in data structures such as lists, tuples, and sets.
- To equip students with modular programming, file handling, exception handling, and basic object-oriented programming skills.

Course Outcomes (COs):

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

CO1: Apply algorithms, flowcharts, variables, data types, and operators to design and implement basic Python programs for solving simple computational problems. (L3)

CO2: Apply conditional and iterative control structures, including if-else, nested if, match, while, and for loops, to develop solutions for decision-making and repetitive problems. (L3)

CO3: Analyze string operations and Python data structures such as lists, tuples, and sets to manipulate, organize, and solve data-oriented problems. (L4)

CO4: Evaluate computational problems and develop modular Python solutions using user-defined functions, recursion, lambda expressions, and dictionaries. (L5)

CO5: Apply file handling, exception handling, and basic object-oriented programming concepts to develop reliable and maintainable Python applications. (L3)

CO – PO Articulation Matrix:

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

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

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 (and, or, not), 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 from the user; 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 ≥ 90, A+ ≥ 80, A ≥ 70, B+ ≥ 60, B ≥ 50, C ≥ 40, F < 40); print a formatted result card using print() with appropriate spacing; (c) accept two numbers and an operator (+, −, ×, ÷) as input; evaluate the expression using eval(); handle division by zero using a simple if condition; (d) demonstrate operator precedence: compute and print results of 5 mixed expressions — predict the result manually first, 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 the following five patterns using nested for loops: right-angled triangle (stars), inverted triangle, number pyramid, Floyd's triangle, and a diamond pattern — accept the number of rows as input for each; (d) implement a number guessing game — generate a random number between 1 and 50 using random.randint(); allow the user a maximum of 7 attempts; print 'Too High' or 'Too Low' as hints; use break when the correct number is guessed; display attempts used.
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 (1+2+…+N), sum of squares ($1^2+2^2+\ldots+N^2$), and sum of cubes ($1^3+2^3+\ldots+N^3$) — 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 on the same index values; (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 given 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 (A | B), intersection (A & B), difference (A − B and B − A), symmetric difference (A ^ B); verify De Morgan's laws: not(A | B) == (not A) & (not B) 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 the 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 (def greet(name, age)), default (def greet(name, age=18)), keyword (greet(age=20, name='Ram')), 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 that is modified inside a function using the global keyword; contrast with a local variable of the same name — print both; (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, Value Error (non-numeric input), and Type Error in separate except blocks; use a finally block to print 'Operation complete' regardless of outcome; (b) implement a robust file reader that handles File Not Found Error and Permission Error; 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 Insufficient Funds Error(Exception) with a message attribute; implement a Bank Account class with deposit() and withdraw() methods — withdraw() raises Insufficient Funds Error if balance is insufficient; test with deposits and withdrawals; (d) demonstrate exception chaining (raise X from Y) — catch a Value Error and re-raise as a custom Application Error 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) demonstrate object interactions: write a find_topper(students) function that accepts a list of Student objects and returns the one with the highest percentage; sort the list of Student objects by percentage using sorted() with a lambda key on obj.compute_percentage(); (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.

Text Books:

1. Pieter Spronck, Computational Thinking with Python (course textbook — algorithms and Python foundations).
2. Reema Thareja, Programming in Python, Oxford University Press. (Primary implementation reference — all five units.)

Reference Books:

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

Online Tools and Resources

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

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES03102P	ES	**Engineering Workshop** (Common to all branches of Engineering)	0	0	0	3	1.5

Pre-Requisites: Basic Computing Elements.

Course Objectives:

- To create awareness on safety precautions required at workplace.
- To master fundamental hand tools and traditional manufacturing craftsmanship.
- To practice on manufacturing of components using workshop trades including fitting, Sheet metal and carpentry, foundry and welding.
- To import basic electrical engineering knowledge for House Wiring Practice.
- To demonstrate fundamentals of foundry and welding.

Course Outcomes (COs):

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

CO1: Apply appropriate safety precautions and procedures while performing various workshop operations. (L3)

CO2: Analyze the selection, application, and operational capabilities of measuring, marking, and cutting tools used in workshop practices. (L4)

CO3: Evaluate and fabricate precision wood and metal fitting joints according to standard specifications and quality requirements. (L5)

CO4: Analyze wire standards, residential switching arrangements, lighting circuits, and basic electrical appliances to ensure safe and effective operation. (L4)

CO5: Evaluate the suitability and application of tools used in foundry and welding sections for different workshop operations. (L5)

CO – PO Articulation Matrix:

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

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

1. Safety Practices & Precautions: Demonstrate the necessity of Workshop hazards, various Personal Protective Equipment and emergency protocols.

AI Integration: Computer vision for safety compliance.

2. Wood working: Introduction to different types of woods, carpentry tools, timber joints

a) Half – Lap joint b) Mortise and Ten on joint c) Corner Dovetail joint or Bridle joint

3. Fitting: Familiarity with different types of tools used in fitting and perform fitting exercises like 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, Developments of following sheet metal job from GI sheets.

a) Tapered tray b) Conical funnel c) Elbow pipe d) Brazing

5. Electrical wiring: Introduction to electrical tools, safety precautions, wiring materials.

Familiarity with different types of basic electrical circuits and make the following connections.

a) Parallel and series b) Two-way switch c) God own lighting d) Tube light e) Three phase motor f) Soldering of wires

6. Plumbing: Introduction to plumbing tools, pipe fittings, thread cutting, making a simple PVC pipe joint. Preparation of Pipe joints with coupling for same diameter and with reducer for different diameters.
7. Welding Shop: Demonstration and practice on Arc Welding and Gas welding. Preparation of a simple Lap joint/Butt joint.
8. Foundry Trade: Demonstration and practice on Moulding tools and processes, Preparation of Green Sand Moulds for a simple Patterns.
9. Introduction to 3D Printing: Demonstrate the core concepts of 3D printing, Hardware, software, Digital file and 3D printing materials. Familiarise with 3D Printing Process of complex objects.
10. Computer Components: Disassemble and reassemble a desktop computer to understand the role of each component (CPU, motherboard, RAM, HDD/SSD, GPU, etc.).
Hardware Troubleshooting & Peripheral Installation: Simulate common hardware issues (e.g., RAM failure, CPU overheating) and identify the procedures to resolving them.
Peripheral Installation: Install and configure peripheral devices such as printers, scanners, and external hard drives.
Network Setup and Configuration: Set up a small LAN network with routers, switches, and computers. Configure IP addresses, subnet masks, and basic network services.
11. Disassembly and Assembly of equipment such as Bicycle, Gear Box, Braking System: Familiarise with disassembly and assembly process
AI Integration Module: Smart energy monitoring using IoT current sensors; building machine learning load forecasting models; training anomaly detection systems to automatically flag electrical arc or leakage patterns. Develop predictive load-monitoring models that detect faulty configurations or phantom energy drains.
Text Books:
1. **Workshop Core Reference**: Workship Technology Vol I & II by S.K. Hajra Choudhury, Media Promoters & Publishers.
2. **Workshop Core Reference**: A Course in Workshop Technology by B.S. Raghuwanshi, Dhanpat Rai & Co. 2015 & 2017
3. **Data-Driven Industrial IoT**: Hands-On Industrial Internet of Things by Giacomo Veneri and Antonio Capasso, Packt Publishing.
4. **Basic Workshop Technology**: Manufacturing Process, Felix W.; Independently Published,2019. Workshop Processes, Practices and Materials; Bruce J. Black, Routledge publishers, 5th Edn. 2015.
5. Wiring Estimating, Costing and Contracting; Soni P.M. & Upadhyay P.A.; Atul Prakashan, 2021-22.

Reference Books:
1. Manufacturing Technology (Foundry, Forming and Welding) by P.N. Rao, McGraw Hill.
2. An Introduction to Predictive Maintenance by R. Keith Mobley, Butterworth-Heinemann

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

B.Tech. – I Year I Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26HMHS103L	HM	**NSS / NCC / Scouts & Guides / Community Service** (Common to all branches of Engineering)	0	0	0	1	0.5

Pre-Requisites: Basic awareness of civic responsibilities, community service, teamwork, discipline, and general social and environmental issues.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the principles and practices of NSS, NCC, Scouts & Guides, and community service to demonstrate responsible citizenship, civic sense, volunteerism, ethical conduct, and active participation in community-development activities. (L3)

CO2: Apply leadership, communication, teamwork, field-craft, camping, disaster-management, and first-aid skills to effectively respond to community, camp, and emergency situations. (L3)

CO3: Apply community needs-assessment, project-planning, social-awareness, environmental-conservation, digital-literacy, and documentation skills to implement and evaluate community service and service-learning activities. (L3)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1						3	2	2	2		2		
CO2						3	3	3	2		2		
CO3						3	2	3	2		2		

Unit I: Foundation of NSS, NCC, Scouts & Guides and Community Service:

- 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 II: **Leadership, Skill Development, Camping and Disaster Management:**

- 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 III: Community Projects, Social Awareness and Service Documentation:

- 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

Text Books:

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:

1. Community Development & Social Service, Dr. S. K. Lal, Regal Publications, 2019.
2. Disaster Management – A Holistic Approach, Satish Modh, Macmillan India, 2020.
3. Volunteer Management, Steve McCurley & Rick Lynch, Energize Publications, 2018.
4. Scouting for Boys, Lord Robert Baden-Powell, Oxford University Press (Centenary Ed.), 2008

Web Resources:

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

YouTube Resources:

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

AI-Integrated Activities:

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

B.Tech. – I Year
II Semester

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB102T	BS	**Differential Equations & Vector Calculus** (Common for all branches of Engineering)	3	0	2	0	4

Pre-Requisites: Linear Algebra and Calculus

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Solve first-order ODEs and model engineering processes using analytical and computational approaches. (L3)

CO2: Analyse higher-order linear ODEs and interpret solutions in dynamic engineering systems such as circuits and oscillators. (L4)

CO3: Formulate and solve first-order and selected higher-order PDEs relevant to engineering models including wave, heat, and Laplace equations. (L3)

CO4: Interpret and compute gradient, divergence, and curl of scalar and vector fields and explain their physical significance. (L4)

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

CO – PO Articulation Matrix:

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

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

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; Engineering applications of first-order ODEs; 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; Classification as hyperbolic, parabolic, and elliptic. Second-order PDE applications — wave equation, heat equation, and Laplace's equation by method of separation of variables.

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

Free/Open-Source AI Tools: Python (SymPy, NumPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; Wolfram Alpha free tier for PDE classification verification; Scilab.

Unit IV: Vector Differentiation:

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

AI Integration: Python NumPy and Matplotlib for 2D and 3D gradient, divergence, and curl field visualization (quiver plots, level-surface overlays); SymPy for symbolic verification of vector identities; 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.

Text Books:

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:

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

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB107T	BS	Chemistry for Civil Engineers	3	0	0	0	3

Pre-Requisites: Basic Chemistry (Class XII level), and Engineering Mathematics

Course Objectives:

- To understand water chemistry parameters relevant to civil engineering applications and to design and evaluate water and wastewater treatment processes.
- To understand electrochemical principles governing corrosion in civil engineering structures and to evaluate corrosion prevention and protection methods for infrastructure applications.
- To understand the chemical composition and hydration reactions of cement and concrete and to analyze the role of admixtures and their chemical effects on concrete properties.
- To understand the chemistry, properties, and applications of polymers in civil engineering and to analyze composite materials used in modern construction.
- To understand principles of green chemistry for sustainable civil engineering practices and to evaluate environmental impacts of construction materials and processes.

Course Outcomes (COs):

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

CO1: Recall and explain water chemistry parameters including hardness, alkalinity, BOD, COD and dissolved oxygen; analyse water quality with reference to civil engineering applications; and apply appropriate internal, external and domestic treatment methods for boiler feed water and potable water supply. (L3)

CO2: Analyze electrochemical principles (electrode potential, Nernst equation, Galvanic series) governing corrosion, identify types of corrosion and their mechanisms in reinforced concrete and steel structures, and evaluate cathodic protection, protective coating, and corrosion inhibitor strategies for civil infrastructure. (L4)

CO3: Explain the chemical composition and hydration reactions of Portland cement (C_3S, C_2S, C_3A, C_4AF), analyze the chemistry of concrete including admixtures and lime-gypsum materials, and select appropriate special cement types and heat-treated steel based on specific construction requirements. (L3)

CO4: Explain the chemical composition and hydration reactions of Portland cement (C_3S, C_2S, C_3A, C_4AF), analyze the chemistry of concrete including admixtures and lime-gypsum materials, and select appropriate special cement types and heat-treated steel based on specific construction requirements. (L4)

CO5: Apply the twelve principles of green chemistry to assess environmental impacts of construction materials and processes; evaluate sustainable alternatives including geopolymer concrete, fly ash, and GGBS; and design strategies to minimize air, water, and soil pollution and manage solid waste in civil engineering projects. (L5)

CO – PO Articulation Matrix:

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

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

Unit I: Water Chemistry and Treatment:

Core Content:

- Water chemistry: hardness of water, temporary and permanent hardness of water, hardness units, problems, estimation of hardness of water by EDTA method
- Boiler troubles: scale and sludge formation, boiler corrosion, caustic embrittlement
- Internal treatment of boiler feed water. a. phosphate conditioning b. Calgon conditioning c. treatment with sodium aluminate d. electrical conditioning e. radioactive conditioning f. complexometric method
- External treatment of boiler feed water- ion exchange, reverse osmosis
- Domestic treatment of water- chlorination, Bleaching powder
- Analysis of water: alkalinity, pH, dissolved oxygen, BOD, COD
- Water treatment processes: screening, sedimentation, coagulation-flocculation, filtration, disinfection
- Chemistry of coagulation: alum, ferric chloride, polyelectrolytes.
- Disinfection: chlorination chemistry, breakpoint chlorination, ozonation, UV treatment
- Wastewater treatment: primary, secondary (activated sludge, trickling filters), tertiary treatment
- Sludge treatment and disposal

AI Integration: Prediction of water quality trends using Python (Pandas, Matplotlib). Regression models for forecasting BOD/COD. Visualization of water treatment efficiency using datasets.

Free/Open-Source AI Tools: Python (NumPy, Pandas, Matplotlib), Google Colab

Unit II: Electrochemistry and Corrosion:

Core Content:

- Electrochemical principles: Electrode potential, Nernst equation, Galvanic series
- Conductance- conductometric titrations a. Strong Acid Vs Strong Base, b. Weak Acid Vs Strong Base.
- Types of corrosion: Chemical Corrosion, pilling bed worth rule and Electro Chemical Corrosion, Concentration cell Corrosion, uniform, pitting, crevice, galvanic, stress corrosion.
- Factors effecting the corrosion.
- Corrosion prevention methods: cathodic protection-Sacrificial anodes and impressed current systems.
- protective coatings-Galvanizing and electroplating for steel protection, electroless plating, tinning and metal cladding.
- corrosion inhibitors.

AI Integration: Machine learning models for corrosion rate prediction using open datasets. Data-driven analysis of environmental corrosion factors.

Free/Open-Source AI Tools: Python (scikit-learn), Google Colab, Kaggle datasets

Unit III: Chemistry of Construction Materials:

Core Content:

- Chemistry of cement: Portland cement composition (C_3S, C_2S, C_3A, C_4AF), hydration reactions, heat of hydration.
- Setting and hardening mechanisms, calcium silicate hydrates (C-S-H gel)
- Special Types of cement and their applications. A. High alumina cements B. High early strength cements (HES) C. Silicate cements D. Sorel cements
- Chemistry of Concrete: Role of aggregates, water-cement ratio, workability.

- Chemical Admixtures: Plasticizers, retarders, accelerators, air-entraining agents.
- Chemistry of lime, gypsum, and clay materials.
- Steel chemistry: composition and applications of steel, alloying elements
- Heat Treatment: annealing, normalizing, hardening, tempering, age hardening.

AI Integration: Prediction of concrete compressive strength using machine learning models. Data analysis of mix design parameters.

Free/Open-Source AI Tools: TensorFlow, Orange Data Mining, Python

Unit IV: Engineering Polymers and Composite Materials:

Core Content:

- Polymer Chemistry: Polymerization mechanisms (addition, condensation), classification (thermoplastics, thermosets)
- Engineering Polymers: Polyethylene, PVC, polypropylene, epoxy resins, polyurethane
- Elastomers: Natural rubber, vulcanisation, synthetic rubbers (BUNA-S and BUNA-N) Applications in civil engineering.
- Composite Materials: Matrix and reinforcement, types (polymer matrix, metal matrix, ceramic matrix)
- Fiber-reinforced polymers (FRP): Glass fiber, carbon fiber, aramid fiber and applications.

AI Integration: Computer vision for polymer structure identification and degradation analysis. Image-based classification of materials.

Free/Open-Source AI Tools: KNIME, Teachable Machine, OpenCV

Unit V: Green Chemistry and Environmental Chemistry:

Core Content:

- Principles of green chemistry and their applications
- Sustainable construction materials: Geopolymer concrete, recycled aggregates, industrial by-products (fly ash, GGBS)
- Air pollution: Types, sources, effects; air quality index
- Water pollution: Sources, effects and remedial methods.
- Chemistry of greenhouse gases and climate change
- Soil chemistry: pH, organic matter, contamination, remediation
- Solid waste management: Classification, treatment, recycling in construction.

AI Integration: Life Cycle Analysis (LCA) using AI tools. Air quality prediction models using machine learning. Sustainability assessment of materials.

Free/Open-Source AI Tools: OpenLCA, Google Colab, Python

Text Books:

1. Jain, P. C., & Jain, M. (2020). Engineering chemistry (17th ed.). Dhanpat Rai Publishing Company.
2. Shikha, A., & Sharma, R. K. (2019). Engineering chemistry. Krishna Prakashan Media.

Reference Books:

1. Davis, M. L. (2020). Water and wastewater engineering: Design principles and practice (2nd ed.). McGraw-Hill Education.
2. Shikha, A., & Sharma, R. K. (2019). Engineering chemistry. Krishna Prakashan Media.
3. Dara, S. S., &Umare, S. S. (2018). Engineering chemistry (15th ed.). S. Chand Publishing.

Web References:

1. American Concrete Institute. (2021). ACI 222R-01: Protection of metals in concrete against corrosion. https://www.concrete.org
2. Bureau of Indian Standards. (2012). IS 456:2000 – Plain and reinforced concrete – Code of practice. https://www.services.bis.gov.in
3. U.S. Environmental Protection Agency. (2023). Green chemistry. https://www.epa.gov/greenchemistry
4. World Health Organization. (2022). Guidelines for drinking-water quality (4th ed.). https://www.who.int/publications/i/item/9789241549950
5. Indian Green Building Council. (2023). LEED India rating systems. https://igbc.in/igbc/

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26HMHS101T	HM	**Communicative English** (Common for all branches of Engineering)	2	0	0	0	2

Pre-Requisites: Basic English (Intermediate level).

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the principles of communication, grammatical structures, tenses, subject–verb agreement, phonetic patterns, and vocabulary skills to identify and correct language errors and demonstrate effective spoken and written communication using AI-assisted tools. (L3)

CO2: Apply effective listening, speaking, group discussion, presentation, and debate techniques to organize and deliver clear, confident, and persuasive oral communication using AI-assisted tools for speech analysis, transcription, and visual presentation design. (L3)

CO3: Apply reading and writing strategies to comprehend, organize, and produce clear paragraphs, creative content, movie reviews, and blogs using AI-assisted paraphrasing, readability, simplification, and content-structuring tools. (L3)

CO4: Analyze professional and business communication requirements to evaluate and improve the structure, tone, formality, and effectiveness of emails, letters, reports, summaries, resumes, and LinkedIn profiles using AI-assisted communication tools. (L4)

CO5: Evaluate the effectiveness, accuracy, and ethical implications of AI-assisted language learning and communication tools, with regard to translation, plagiarism, academic integrity, bias, privacy, misinformation, and responsible digital communication. (L5)

CO – PO Articulation Matrix:

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

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

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)

Text Books:

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:

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

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES03103T	ES	**Design Thinking** (Common for all branches of Engineering)	1	0	4	0	3

Pre-Requisites: Basic knowledge of problem-solving, creativity, communication, teamwork, and fundamental engineering concepts.

Course Objectives:

- To understand the principles, philosophy, stages, and importance of Design Thinking in human-centred engineering problem solving.
- To apply empathy methods such as interviews, observation, personas, and problem-framing techniques to identify and understand user needs.
- To analyse user needs, formulate effective problem statements using the How Might We framework, and leverage AI tools for generating insights and evaluating solutions.
- To develop, test, evaluate, and pitch innovative prototype solutions using AI ideation tools, user feedback, and digital prototyping platforms.

Course Outcomes (COs):

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

CO1: Recall and understand the philosophy, principles, and stages of the Design Thinking process and its relevance to human-centred problem solving. (L3)

CO2: Apply empathy tools (interviews, observation, personas) and problem-framing techniques to real-world engineering scenarios. (L3)

CO3: Analyse user needs and problem statements using the 'How Might We' framework and AI-powered insight generation tools. (L4)

CO4: Evaluate and select the most viable prototype solution through structured testing, user feedback, and AI-assisted evaluation rubrics. (L5)

CO5: Create innovative prototypes and pitch solutions to identified problems using AI ideation tools and digital prototyping platforms. (L6)

CO – PO Articulation Matrix:

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

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

Unit 1: Introduction to Design Thinking:

Core Concepts:

- Design Thinking: Definition, origin (Stanford d.school, IDEO), and relevance to engineering
- The 5-Stage Model: Empathise – Define – Ideate – Prototype – Test
- Human-centred design principles: Desirability, Feasibility, Viability
- Systems Thinking vs. Linear Thinking: Understanding complexity
- Case Studies: Successful design thinking applications (IDEO shopping cart, Stanford Mobile Clinic)

AI Integration:

- AI Tool: ChatGPT / Copilot – use AI to explain design thinking in different domains (healthcare, education, agriculture)

- Activity: Use MindMeister (free, open-source alternative: FreeMind) to create a mindmap of the 5-stage process
- AI Video Analysis: Watch an IDEO documentary and use AI to summarise key DT principles observed

Unit II: Empathise and Define Stages:

Core Concepts:

- Empathy Methods: Interviews, Observation, Immersion, Shadowing
- Creating User Personas: Demographic, psychographic, goals and pain points
- Empathy Mapping: Says, Thinks, Does, Feels
- Point of View (POV) Statement: User + Need + Insight formula
- How Might We (HMW) Questions: Reframing problems as opportunities

AI Integration:

- AI Tool: Miro (free plan) – create digital empathy maps and POV statements collaboratively
- AI Tool: ChatGPT – generate diverse user persona profiles for a given problem context
- Activity: Interview 5 peers about a campus problem; use AI to analyse interview transcripts and extract themes
- AI Tool: Dovetail (free plan) – affinity mapping of qualitative research data

Unit III: Ideation Stage:

Core Concepts:

- Creative Thinking Techniques: Brainstorming, Brainwriting, SCAMPER, Random Word Association
- Lateral Thinking: Edward de Bono's Six Thinking Hats
- Affinity Diagramming: Grouping ideas by themes
- Prioritisation Matrix: Impact vs. Effort, Dot Voting
- Concept Selection: Morphological charts and Pugh Matrix

AI Integration:

- AI Tool: ChatGPT – use 'expand my idea' prompts to generate 20 variations of a concept
- AI Tool: IdeaFlip (free) / Miro – digital brainstorming and affinity diagramming
- AI Tool: Midjourney (free trial) / DALL-E (free credits) – generate visual concept sketches from text descriptions
- Activity: Run a 30-minute AI-augmented brainstorming session and compare idea quality with pure human brainstorming

Unit IV: Prototyping:

Core Concepts:

- Principles of Prototyping: Low-fidelity vs. High-fidelity prototypes
- Paper Prototyping: Sketching UI and physical models
- Digital Wireframing: Introduction to wireframes and mockups
- 3D Prototyping Concepts: Introduction to 3D printing and rapid prototyping
- Role Plays and Experience Prototyping: Service design scenarios

AI Integration:

- AI Tool: Figma (free plan) – create digital wireframes and interactive mockups
- AI Tool: Uizard (free plan, AI-powered) – convert hand-drawn sketches to digital wireframes using AI
- AI Tool: Tinkercad (free, web-based) – create simple 3D models for physical prototyping
- Activity: Build a paper prototype for a chosen problem and photograph for digital documentation

Unit V: Testing, Iteration and AI in Design:

Core Concepts:

- User Testing Methods: Think-aloud protocol, A/B testing, Usability testing
- Feedback Analysis: Affinity mapping of test feedback
- Iteration Cycle: Incorporating feedback and pivoting
- Design Presentation: Pitching to stakeholders (elevator pitch, demo day format)
- AI in Design: Overview of AI-assisted design (generative design, co-creation with AI)

AI Integration:

- AI Tool: Maze (free plan) – conduct remote usability tests on Figma prototypes
- AI Tool: ChatGPT – analyse collected user feedback and generate insights and improvement suggestions
- Activity: Present a final prototype to a panel; use AI to prepare pitch slides in Canva
- Reflection: Create a Design Thinking journal documenting each stage with AI tool usage notes

Text Books:

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

Reference Books:

1. IDEO.org. (2015). The Field Guide to Human-Centered Design. IDEO.org, San Francisco.
2. Kelley, Tom & Kelley, David. (2013). Creative Confidence: Unleashing the Creative Potential Within Us All. Crown Business.
3. Liedtka, Jeanne & Ogilvie, Tim. (2011). Designing for Growth: A Design Thinking Tool Kit for Managers. Columbia Business School Press.
4. Plattner, Hasso, Meinel, Christoph & Leifer, Larry (Eds.). (2018). Design Thinking Research. Springer.
5. Kumar, Vijay. (2012). 101 Design Methods: A Structured Approach for Driving Innovation. Wiley.
6. Stickdorn, Marc & Schneider, Jakob. (2021). This Is Service Design Thinking (Expanded ed.). BIS Publishers.

Web Resources:

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

YouTube / Video Resources:

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

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26PC01101T	PC	Engineering Mechanics	2	0	2	0	3

Pre-Requisites: Engineering Mathematics (Linear Algebra & Calculus, Differential Equations).

Course Objectives:

- To introduce fundamental concepts of mechanics, force systems, and equilibrium with applications in structural and mechanical engineering.
- To develop ability to draw and interpret free body diagrams for coplanar, concurrent, and spatial force systems.
- To calculate centroid, centre of gravity, and moment of inertia for simple and composite bodies.
- To understand the fundamental concepts of stress, strain, elasticity, and the mechanical behaviour of materials under different loading conditions.
- To introduce AI/ML tools for simulation, analysis, and visualization of mechanical systems.

Course Outcomes (COs):

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

CO1: Apply the fundamental principles of mechanics, including Newton's laws, types of forces, equilibrium conditions, and Coulomb's laws of friction, to solve basic problems and demonstrate their applications in Civil and Mechanical Engineering. (L3)

CO2: Apply the principles of force systems, free body diagrams, and equilibrium equations to solve coplanar, concurrent, and spatial force problems including trusses, frames, and friction-based mechanisms using analytical and computational methods. (L3)

CO3: Analyze the geometric properties of engineering sections and bodies by calculating centroids, centre of gravity, and moments of inertia for composite shapes, and evaluate their significance in structural design using software tools. (L4)

CO4: Evaluate the motion and dynamic response of engineering systems using kinematics and kinetics of particles, applying work-energy and impulse-momentum principles and comparing analytical solutions with AI-based simulation outputs. (L5)

CO5: Apply the fundamental concepts of stress, strain, Hooke's law, elastic constants, and their inter-relationships to analyze the mechanical behaviour of engineering materials under axial, thermal, and composite loading conditions. (L3)

CO – PO Articulation Matrix:

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

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

Unit I: Introduction, Force Systems & Friction:

Core Content: Basic concepts of mechanics: scalar and vector quantities, Newton's laws, SI units. Coplanar concurrent forces: resultant by parallelogram law, triangle law, polygon law. Moment of a force: Varignon's theorem. Couples and resultants of force systems. Lami's theorem. Spatial force systems: vector representation, dot and cross products. Friction: limiting friction, Coulomb's laws of dry friction, coefficient of friction, angle of friction, cone of static friction, impending motion problems, wedge friction, screw jack.

AI Integration: Introduction to AI in Engineering: concept of data, machine learning (ML), and simulation. Using Python (NumPy) to solve force equilibrium problems computationally. Introduction to MATLAB/Scilab for vector force calculations. Visualizing force systems using Matplotlib/Python.

Free/Open-Source AI Tools: Python (NumPy, Matplotlib) – Google Colab (free); Scilab (free MATLAB alternative); GeoGebra (free online vector visualization)

Unit II: Equilibrium of Force Systems & Virtual Work:

Core Content: Equilibrium conditions for coplanar systems: free body diagrams, equations of equilibrium ($\Sigma Fx=0$, $\Sigma Fy=0$, $\Sigma M=0$). Graphical methods: triangle and polygon of forces. Equilibrium of spatial systems. Analysis of plane trusses: method of joints, method of sections. Analysis of frames. Principle of virtual work: statement, applications to simple mechanisms and trusses.

AI Integration: Using Python/OpenSees (open source) for automated truss analysis. AI-based structural analysis: introduction to how FEM software uses matrix methods internally. Computer simulation of mechanism motion using Python. Visual debugging of FBDs using interactive tools.

Free/Open-Source AI Tools: OpenSees (free, open-source structural analysis); Python scipy. optimize for equilibrium solution; SAP2000 student edition (free); online truss solvers (free)

Unit III: Centroid, Centre of Gravity & Moment of Inertia:

Core Content: Centroid of lines, areas, and volumes from first principles. Centroid of standard shapes: rectangle, triangle, circle, semicircle, quarter circle. Centroid of composite figures using tabular method. Centre of gravity of 3D bodies: prismatic bodies, cones, hemispheres, composite bodies. Pappus theorems and their applications. Area moment of inertia: polar moment, transfer theorem (parallel axis theorem). MOI of composite sections. Product of inertia. Mass moment of inertia for simple and composite bodies.

AI Integration: Python scripting for automated centroid and MOI calculations of composite cross-sections. Using spreadsheets (LibreOffice Calc) and Python Pandas for tabular calculations. Introduction to how CAD software (AutoCAD/Free CAD) computes section properties. Application in AI-driven structural design optimization: cross-section selection.

Free/Open-Source AI Tools: Python (NumPy, Pandas) – Google Colab; Free CAD (free, open-source CAD with section property computation); LibreOffice Calc (free)

Unit IV: Kinematics & Kinetics of Particles:

Core Content: Kinematics: rectilinear motion – displacement, velocity, acceleration relations; curvilinear motion – Cartesian and polar coordinates; projectile motion. Kinetics: Newton's second law; D'Alembert's principle; work-energy method: kinetic energy, work done, power, efficiency; impulse-momentum method: linear and angular; impact: direct and oblique central impact, coefficient of restitution.

AI Integration: AI in Autonomous Vehicles: how kinematic equations underpin self-driving car algorithms. Python simulation of projectile motion with drag (real-world). ML-based trajectory prediction: introduction to LSTM for motion prediction. Robot path planning using kinematic constraints. Using Python (matplotlib. animation) for interactive motion visualization.

Free/Open-Source AI Tools: Python (NumPy, Matplotlib, SciPy) – Google Colab; Tracker (free video analysis software for kinematics experiments); Algodoo (free 2D physics simulation)

Unit V: Simple Stresses, Strains & Elastic Constants:

Core Content: Elasticity and plasticity; types of stresses (tensile, compressive, shear) and strains (linear, lateral, shear, volumetric); Hooke's law; Poisson's ratio; elastic constants: E, G, K and their relationships; factor of safety; bars of varying cross-section; elongation under self-weight; bars in series and parallel; temperature stresses; stresses in composite bars. Introduction to principal stresses and strains: principal planes, Mohr's circle for 2D stress state. **Strain Energy** Resilience–Gradual, sudden, impact and shock loadings–simple applications.

AI Integration: ML for material property prediction: using regression models to predict Young's modulus and Poisson's ratio from material composition data. Python-based stress-strain curve fitting using SciPy. Introduction to data-driven material models in structural analysis. Overview of physics-informed neural networks (PINNs) for material mechanics (conceptual).

Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib) on Google Colab; Scilab (free); MIT Open Course Ware stress analysis tools

Text Books:

1. Timoshenko S., Young D.H., Rao J.V. and Pati S., Engineering Mechanics, McGraw Hill Education, 5th Edition, 2017.
2. Bhavikatti S.S., A Textbook of Engineering Mechanics, New Age International Publications, 4th Edition, 2018.
3. Timoshenko S. and Gere J., Mechanics of Materials, CBS Publishers, 2006.
4. Strength of materials, R.K.Rajput, S.Chand&Co, NewDelhi
5. Strength of Materials, Basavarajaiah B.S. and Mahadevappa P., Universities Press, 3rd Edition, 2010

Reference Books:

1. Hibbeler R.C., Engineering Mechanics: Statics and Dynamics, Pearson Education, 14th Edition, 2022.
2. Meriam J.L. and Kraige L.G., Engineering Mechanics (Statics & Dynamics), John Wiley, 6th Edition, 2008.
3. Shames I.H., Engineering Mechanics: Statics and Dynamics, PHI, 4th Edition, 2002.
4. Beer F.P. and Johnston E.R., Vector Mechanics for Engineers (Statics & Dynamics), McGraw Hill, 12th Edition, 2019.
5. Bhattacharya B., Introduction to Statics and Dynamics, Oxford University Press, 2nd Edition, 2014.
6. Goldberg D.E., Schaum's Outline of Fluid Mechanics and Hydraulics, McGraw Hill, 4th Edition, 2014.

Web Resources & NPTEL Links:

1. NPTEL: Engineering Mechanics – https://nptel.ac.in/courses/112105171
2. NPTEL: Rigid Body Dynamics – https://nptel.ac.in/courses/112104120
3. PhET Simulations (University of Colorado, Free): https://phet.colorado.edu/en/simulations
4. MIT OpenCourseWare – Classical Mechanics: https://ocw.mit.edu/courses/8-01sc-classical-mechanics
5. Google Colab for Python: https://colab.research.google.com
6. OpenSees documentation: https://opensees.berkeley.edu
7. Tracker Video Analysis (Free): https://physlets.org/tracker

Recommended Free & Open-Source AI/Computational Tool:

1. Python (NumPy, SciPy, Matplotlib, scikit-learn) – Available FREE on Google Colab
2. Scilab 6.x – Free, open-source MATLAB alternative for numerical computation (scilab.org)

3. OpenSees – Free, open-source structural/earthquake engineering simulation framework
4. FreeCAD – Free, open-source 3D CAD with section property computation (freecadweb.org)
5. Tracker – Free video-based physics analysis tool for kinematics experiments (physlets.org/tracker)
6. GeoGebra – Free interactive mathematics and physics visualization (geogebra.org)
7. Algodoo – Free 2D physics simulation environment (algodoo.com)

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26BSHB107P	BS	Chemistry for Civil Engineers Lab	0	0	0	3	1.5

Pre-Requisites: Basic knowledge of general chemistry, chemical reactions, solution preparation, concentration calculations, and fundamental laboratory safety and measurement techniques.

Course Objectives:

- To develop fundamental knowledge of chemical analysis techniques, including volumetric, redox, complexometric, acid-base, and iodometric titrations.
- To develop practical skills for assessing water quality parameters such as hardness, alkalinity, and dissolved oxygen using standard analytical methods.
- To understand the principles of corrosion and evaluate the influence of environmental conditions on the corrosion behaviour of metals.
- To develop proficiency in instrumental and physicochemical analytical techniques for determining chemical constituents and properties of engineering materials and lubricants.

Course Outcomes (COs):

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

CO1: Apply principles of volumetric, redox, complexometric, acid-base, and iodometric titrations to determine the concentration of ions and chemical constituents in given samples. (L3)

CO2: Analyse water quality parameters such as total, permanent and temporary hardness, alkalinity, and dissolved oxygen using appropriate analytical techniques. (L4)

CO3: Determine corrosion behaviour under different environmental conditions and evaluate the factors influencing corrosion of metals. (L3)

CO4: Apply instrumental analytical techniques such as conductometry, pH-metry, spectrophotometry, and colorimetry to identify and quantify chemical species in laboratory samples. (L3)

CO5: Perform physicochemical analysis of materials, including determination of iron in cement and viscosity of lubricating oil, and interpret experimental results accurately using appropriate laboratory procedures. (L3)

CO – PO Articulation Matrix:

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

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

List of Experiments:

1. Estimation of Ferrous Iron (Fe^{+2}) using potassium dichromate ($K_2Cr_2O_7$) (Redox titrations)
2. Determination of Calcium Ca^{+2} using Ethelene diamine tetra acetic acid (EDTA) (complexometric titrations)
3. Estimation of Total hardness, permanent and temporary hardness of water by using Ethelene diamine tetra acetic acid (EDTA) (complexometric titrations)
4. Determination of Total alkalinity (due to OH^- and CO_3^{2-}) of water using sodium hydroxide (NaOH) (acid-base titration)
5. Estimation of Dissolved Oxygen in water sample. (Iodometry)

6. Determination of corrosion.
7. Determination of corrosion by environment.
8. Conductometric titrations of strong acid Vs strong base (Ex: HCl Vs NaOH)
9. pH metry – Estimation of Hydrochloric acid (HCl) using Sodium Hydroxide (NaOH)
10. Spectrophotometry – Determination of wavelength (λ) and verification of Beer's – Lamberts law.
11. Determination of Iron (Fe) in cement by colorimetry.
12. Determination of viscosity of lubricating oil using Red-Wood Viscometer – I

AI Integration: Data collection, visualization, and analysis using Python/Excel. Correlation analysis (e.g., hardness vs alkalinity).

Web Resources:

- WHO
- EPA
- BIS
- Concrete.org
- IGBC

Recommended AI Tools:

- Python (NumPy, Pandas, Matplotlib)
- TensorFlow / Scikit-learn
- Orange Data Mining
- OpenLCA

B. Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26HMHS101P	HM	**Communicative English Lab** (Common for all branches of Engineering)	0	0	0	2	1

Pre-Requisites: Basic knowledge of English grammar, vocabulary, reading, writing, and everyday communication, with a willingness to participate in listening, speaking, and digital language-learning activities.

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

Course Outcomes (COs):

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

CO1: Pronounce English words correctly using standard phonemic symbols and demonstrate improved speech fluency. (L3)

CO2: Apply active listening skills to comprehend spoken English and respond appropriately in formal and informal communication contexts. (L3)

CO3: Participate effectively in group discussions, presentations, and debates demonstrating logical argumentation. (L4)

CO4: Write grammatically correct and contextually appropriate professional documents including emails, reports, and resumes. (L6)

CO5: Utilize AI-powered tools for language learning, self-evaluation, and continuous improvement of English skills. (L5)

CO – PO Articulation Matrix:

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

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

Lab Syllabus – 12 Topics with AI Integration:

S. No.	Lab Topic	Lab Activities	AI Activity	AI Tools	COs
1.	Phonetics & Pronunciation Practice	Record & playback vowel/consonant articulation; IPA chart drill; Minimal pair exercises	Use AI speech-to-text to detect mispronunciations; real-time phoneme feedback via ELSA Speak	ELSA Speak, Google Speech Recognition, Speechling	CO1 & CO5

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 Letter Writing	ATS-friendly resume design; action verbs usage; tailoring cover letters to job descriptions	AI resume scanner; keyword optimisation suggestions; ChatGPT cover letter generation and critique	Kickresume, Resume.io, Jobscan, ChatGPT	CO4 & 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	AI generates arguments on both sides; evaluate	ChatGPT, Claude.ai, Kialo Edu	CO2 & CO3

		techniques; logical fallacy identification	logical structure using ChatGPT		
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

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.

Reference Books:

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

Online Resources:

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

YouTube Channels:

1. English with Lucy – @EnglishwithLucy (Vocabulary, speaking, British accent)
2. Learn English with TV Series – @LearnEnglishWithTVSeries
3. Rachel's English – @rachelsenglish (American accent and pronunciation)

4. TED-Ed – @TEDEd (Critical thinking and communication ideas)
5. Spoken English Hub – @SpokenEnglishHub (Indian context, interview prep)
6. JenniferESL – @JenniferESL (Grammar and business English)

NPTEL / Swayam Courses:

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

Lab Software & AI Tools Required:

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

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26PC01102P	PC	Civil Engineering Workshop	0	0	0	2	1

Pre-Requisites: Engineering Mathematics (Class XII) and Engineering Mechanics (concurrent)

Course Objectives:

- To perform building setting-out exercises using tape and cross staff as per a given plan.
- To construct brick walls using English bond and Flemish bond as dry masonry demonstrations.
- To carry out plastering, putty application, painting and tile flooring as practical finishing works.
- To install plumbing fixtures and prepare soil cement blocks, fly ash blocks and cover blocks; test for compressive strength.
- To apply digital tools (Python, AutoCAD / Libre CAD) for CG and MOI computation, floor plan drawing and AI-assisted civil engineering tasks.

Course Outcomes (COs):

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

CO1: Set out a single-room building as per a given plan using tape and cross staff; construct dry brick walls of varying thickness (1½ and 2 bricks) using English bond (two types) and Flemish bond; identify the bonding pattern, header and stretcher courses, queen closer positions and corner junction layout. (L3)

CO2: Carry out plastering and finishing of a wall surface; apply wall putty and paint a wall following the standard sequence of base coat, putty and finish coat; apply base coat and lay tile flooring for one square metre following standard tiling procedure; identify tools, materials and quality standards involved. (L3)

CO3: Install basic plumbing fixtures and pipe fittings (tap, T-joint, elbow, bend, threading); prepare soil cement blocks by the standard procedure and cast fly ash blocks; test prepared blocks for compressive strength; prepare cover blocks and explain their function in reinforced concrete construction. (L3)

CO4: Compute the centre of gravity (CG) and moment of inertia (I^{xx}, I^{yy}) of a given rolled steel section (ISMB / ISMC) from actual physical measurements; compare computed values with standard IS section property tables; analyse the effect of section geometry on MOI and its significance in structural design. (L4)

CO5: Compute CG and MOI of the steel section using Python (NumPy) and compare with standard values; draw the building floor plan to scale using AutoCAD / Libre CAD; compute brick quantities for English and Flemish bond walls using a Python / Pandas script; evaluate percentage error between measured/computed and standard values. (L5)

CO – PO Articulation Matrix:

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

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

List of Experiments:

Part A: Regular Experiments:

1. Setting out of a building: The student should set out a building (single room only) as per the given building plan using tape and cross staff
2. Construct a wall of height 50 cm and wall thickness 1½ bricks using English bond (No mortar required) – corner portion – length of side walls 60 cm
3. Construct a wall of height 50 cm and wall thickness 2 bricks using English bond (No mortar required) – corner portion – length of side walls 60 cm
4. Construct a wall of height 50 cm and wall thickness 2 bricks using Flemish bond (No mortar required) – corner portion – length of side walls 60 cm
5. Computation of Centre of gravity and Moment of inertia of a given rolled steel section by actual measurements
6. Installation of plumbing and fixtures like Tap, T-Joint, Elbow, Bend, Threading etc.
7. Plastering and finishing of wall
8. Application of wall putty and painting a wall
9. Application of base coat and laying of Tile flooring of one square meter
10. Preparation of soil cement blocks for masonry and testing for compressive strength
11. Casting and testing of Fly ash Blocks
12. Preparation of cover blocks for providing cover to reinforcement

Part B: AI Integration Lab Exercises:

1. CG and MOI Computation (Python / Google Colab): input the measured dimensions of the rolled steel section from Experiment 5 (flange width, flange thickness, web depth, web thickness); compute centroidal distances, area moments I^{xx} and I^{yy} and radius of gyration using NumPy; compare computed values with ISMB / ISMC standard section property tables and compute percentage error.
2. Digital Floor Plan Drawing (AutoCAD Student Version / Libre CAD – free): using the building plan dimensions from Experiment 1 (setting out), draw the floor plan of the single room to scale (1:20) in AutoCAD student version or Libre CAD; add dimensions, door / window symbols, north arrow and title block; export as PDF.
3. AI-Assisted Building Material Scheduling (Python / Pandas): for the wall construction exercises (Experiments 2–4), input wall dimensions and bond type; write a Python script to compute the number of bricks required per course, per leaf and for the full wall height using the appropriate brick layout formula; compare with a manual count from the demonstration.

AI Integration: AI integration adds three exercises following the workshop exercises: (i) Python computation of CG and MOI for the rolled steel section (Experiment 5) – automating the calculations using NumPy and comparing with standard IS section tables; (ii) floor plan drawing in AutoCAD student version or Libre CAD (free) using the building dimensions from Experiment 1 – introducing the industry-standard CAD environment at the first year level; (iii) Python / Pandas brick quantity estimation script for Experiments 2–4 – computing bricks per course and per wall for English and Flemish bonds, linking the physical masonry demonstration to a digital quantity takeoff – an early introduction to BIM quantity estimation concepts.

Free / Open-Source AI Tools: Python (NumPy, Pandas, Matplotlib) – Google Colab (free); AutoCAD 2024 student version (free – autodesk.com/education) or Libre CAD (free, open-source 2D CAD – librecad.org).

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26HMHS102L	HM	**Health and Wellness, Yoga and Sports** (Common for all branches of Engineering)	0	0	0	1	0.5

Pre-Requisites: Basic knowledge of health, fitness, nutrition, and hygiene; willingness to participate in yoga, meditation, sports, and physical activities; basic digital literacy; and an interest in using technology and AI tools for health, fitness, and wellness management.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Demonstrate healthy lifestyle practices by applying principles of holistic health, balanced nutrition, physical fitness, stress management, sleep hygiene, personal hygiene, disease prevention, and ergonomic safety in daily life. (L3)

CO2: Demonstrate appropriate yoga, pranayama, meditation, mindfulness, and relaxation practices with correct techniques and breathing patterns to enhance flexibility, strength, balance, stress management, and overall well-being. (L3)

CO3: Analyze physical fitness components, training principles, sports skills, injury-prevention strategies, psychological factors, gender-inclusive practices, and sports technologies to evaluate and enhance individual and team performance. (L4)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1						3	3	3	2		2		
CO2						3	3	3	2		2		
CO3						3	3	3	2		2		

Unit I: Health, Wellness and Preventive Healthcare:

- Concept of health, wellness and fitness – definitions and seven dimensions
- WHO definition of health; holistic health model (Physical, Mental, Social, Spiritual)
- Nutrition basics – macronutrients, micronutrients, balanced diet, BMI calculation
- Lifestyle diseases – diabetes, hypertension, obesity – causes and prevention
- Mental health & stress management – causes, symptoms, COPE model
- Sleep hygiene and circadian rhythm regulation
- Substance abuse – awareness, prevention and rehabilitation
- Personal hygiene, sanitation and communicable disease prevention
- Ergonomics and occupational health for students

Unit II: Yoga, Pranayama, Meditation and Mind-Body Practice:

- Introduction to Yoga – history, philosophy, Patanjali's Ashtanga Yoga
- Surya Namaskar – 12-step sequence with correct breathing

- Asanas for flexibility: Tadasana, Trikonasana, Paschimottanasana, Sarvangasana
- Asanas for strength & balance: Virabhadrasana I & II, Vrikshasana, Naukasana
- Pranayama techniques: Anulom-Vilom, Bhastrika, Kapalbhati, Bhramari
- Meditation and mindfulness – Dharana, Dhyana, body-scan, guided relaxation
- Yoga Nidra for sleep and stress relief
- Yoga for specific conditions – stress, back pain, PCOD/PCOS, anxiety
- Shatkarmas (cleansing processes) – an overview; International Day of Yoga

Unit III: Physical Fitness, Sports Skills and Sports Technology:

- Physical fitness components – strength, endurance, flexibility, agility, speed, coordination
- Principles of training
- – overload, progression, specificity, reversibility, individuality
- Warm-up, cool-down and dynamic stretching routines
- Indoor sports: Badminton, Table Tennis, Chess, Carrom – rules and basic skills
- Outdoor sports: Cricket, Football, Volleyball, Kabaddi, Athletics – rules and skills
- Sports injuries – types, RICE method, prevention and return-to-play protocol
- Sports psychology – motivation, goal setting, concentration and visualisation
- Gender equity and inclusivity in sports; Para-sports awareness
- Sports technology – wearables, fitness trackers, GPS analytics, performance

Text Books:

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

Reference Books:

1. Yoga: The Iyengar Way, Silva, Mira & Shyam Mehta, Dorling Kindersley, 2019.
2. Sports Medicine Essentials, Jim Clover, Cengage Learning, 2020.
3. Foundations of Physical Education & Sport, Charles A. Bucher, McGraw-Hill Education, 2018.
4. Mindfulness: An Eight-Week Plan for Finding Peace, Mark Williams & Danny Penman, Rodale Books, 2018.

Web Resources:

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

YouTube Resources:

1. Surya Namaskar – Complete 12-Step Tutorial – https://youtu.be/surya_tut_01
2. Pranayama for Beginners (Swami Ramdev) – https://youtu.be/pranayama_sr_02
3. Mindfulness Meditation (10 Min) – https://youtu.be/mindful_med_03
4. Balanced Nutrition Explained (ICMR) – https://youtu.be/nutrition_icmr_04

5. Sports Injury Prevention Tips – https://youtu.be/injury_prev_05
6. Mental Health Awareness for Students – https://youtu.be/mh_students_06
7. Yoga for Stress Relief – Isha Foundation – https://youtu.be/yoga_isha_07
8. Fit India Movement – PM Modi Launch – https://youtu.be/fitindia_pmo_08

AI-Integrated Activities:

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

B.Tech. – I Year II Semester

Course Code	Category	Name of the Course	L	T	Pr.	P	C
26ES05102A	MC	**Cybersecurity Awareness for Engineers** (Common for all branches of Engineering)	2	0	0	0	0

Pre-Requisites: Basic knowledge of computer systems, networking, internet usage, digital communication, information security, and fundamental cyber safety practices.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply fundamental cybersecurity concepts to address security requirements in digital environments. (L3)

CO2: Analyze common cyber threats, cyberattacks, and social engineering techniques to determine appropriate security measures. (L4)

CO3: Apply secure practices for protecting devices, networks, email, cloud services, and online platforms. (L3)

CO4: Evaluate cyber ethics, privacy protection measures, and applicable cyber laws in different digital scenarios. (L5)

CO5: Evaluate cybersecurity incidents and **develop** appropriate cyber hygiene and response strategies. (L5)

CO – PO Articulation Matrix:

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

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

Unit I: Introduction to Cyberspace and Cybersecurity:

Introduction to cyberspace and cybersecurity – Evolution of information systems and Internet – Importance of cybersecurity in engineering and society – Information security concepts – CIA Triad (Confidentiality, Integrity, Availability) – Digital footprint and digital identity – Security governance basics – Cyber threat actors – Engineering ethics and responsible digital behavior – Cyber hygiene practices.

Unit II: Cyber Threats, Cybercrime and Social Engineering:

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.

Unit III: Personal Cyber Safety and Cyber Security Technologies:
Password management and Multi-Factor Authentication – Authentication and access control concepts – Safe browsing practices – Email and communication security – Mobile device security – Cloud security basics – Secure use of Wi-Fi and public networks – Data backup and recovery – Cryptography and digital signatures (awareness level only) – Safe use of AI tools and digital platforms.

Unit IV: Privacy, Cyber Ethics and Cyber Laws:
Data protection and privacy concepts – Personal data and sensitive information – Cyber ethics and responsible digital behavior – Overview of cyber laws in India – Information Technology (IT) Act and related provisions (awareness level) – Common cybercrimes and legal implications – Cybercrime reporting mechanisms and grievance redressal – Digital evidence basics – Social media responsibility and legal considerations.

Unit V: Cyber Hygiene and Secure Digital Practices:
Cybersecurity risk awareness – Password management and Multi-Factor Authentication – Secure use of email, web, cloud and public networks – Safe digital transactions and digital payment awareness – Secure use of social media and AI tools – Cybersecurity awareness in IoT and emerging technologies – Incident awareness and cyber hygiene best practices.

Text Books:

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

Reference Books:

1. Charles J. Brooks, Christopher Grow, Philip Craig and Donald Short, Cybersecurity Essentials, Wiley
2. William Stallings and Lawrie Brown, Computer Security: Principles and Practice, Pearson.
3. Michael E. Whitman and Herbert J. Mattord, Principles of Information Security, Cengage Learning.
4. Joseph MiggaKizza, Guide to Computer Network Security, Springer.
5. Eric Cole, Cybersecurity for Beginners, Wiley.

Online Resources:

1. National Cyber Security Awareness Portal (India)
2. CERT-In (Indian Computer Emergency Response Team)
3. National Cyber Crime Reporting Portal
4. NIST Cybersecurity Framework Resources
5. OWASP Foundation
6. Cisco Networking Academy – Introduction to Cybersecurity

B. Tech. – II Year
I Semester

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26BSHB211T	BS	**Complex Variables and Probability Theory** (Common to CE, EEE & ECE)	2	1	0	0	3

Pre-Requisites: Basics of Engineering Mathematics, Calculus and Differential Equations

Course Objectives:

- To develop a thorough understanding of complex variable theory, analyticity, Cauchy–Riemann conditions, and harmonic functions as foundations for engineering analysis.
- To apply complex integration techniques, series expansions, and the residue theorem to evaluate contour integrals and real definite integrals arising in engineering problems.
- To formulate probabilistic models using axioms, random variable theory, and standard distributions, and compute correlation and regression measures for data analysis.
- To integrate analytical methods with AI-assisted computational tools for visualization, simulation, and verification of complex and probabilistic computations.
- To develop outcome-oriented problem-solving skills aligned with NEP 2020 and modern engineering applications.

Course Outcomes (COs):

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

CO1: Apply the axioms and theorems of probability to determine probabilities and use the properties of discrete and continuous random variables to calculate expectation and variance. (L3)

CO2: Evaluate probabilities using binomial, Poisson, and normal distributions and interpret correlation and regression relationships in engineering data sets. (L5)

CO3: Recall definitions of limits, continuity, and differentiability of complex functions and state the Cauchy–Riemann equations in Cartesian and polar forms. (L1)

CO4: Apply Cauchy's integral theorem, Cauchy's integral formula, Liouville's theorem, and the maximum modulus theorem to evaluate complex line integrals. (L3)

CO5: Analyse complex functions using Taylor's and Laurent's series, classify singularities, and evaluate real definite integrals by the residue theorem. (L4)

CO – PO Articulation Matrix:

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

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

Unit I: 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 II: 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.

Unit III: 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–Thoms on construction.

Free/Open-Source AI Tools: Python (NumPy, Matplotlib, SymPy) on Google Colab; Jupyter Notebook; WolframAlpha free tier for verification; GNU Octave.

Unit IV: Complex Integration:

Core Content: Line integrals in the complex plane; Cauchy's integral theorem; Cauchy's integral formula; Generalized 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 V: 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.

Text Books:

1. E. Kreyszig,Advanced Engineering Mathematics, 10th Edition, Wiley, 2011.
2. R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye,Probability& Statistics for Engineers and Scientists, 9th Edition, Pearson, 2016.
3. R. V. Churchill and J. W. Brown,Complex Variables and Applications, 9th Edition, McGraw-Hill, 2013.
4. J. Miller and J. E. Freund,Probability and Statistics for Engineers, 9th Edition, Pearson, 2014.
5. R. K. Jain and S. R. K. Iyengar,Advanced Engineering Mathematics, 5th Edition, Alpha Science International, 2021.

Reference Books:

1. S. M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, 5th Edition, Academic Press, 2014.
2. B. S. Grewal, Higher Engineering Mathematics, 44th Edition, Khanna Publishers, 2021.
3. Dennis G. Zill,A First Course in Complex Analysis with Applications, 3rd Edition, Jones & Bartlett,

2018.

4. S. C. Gupta and V. K. Kapoor, Fundamentals of Mathematical Statistics, 12th Edition, Sultan Chand & Sons, 2020.

Online Resources:

1. NPTEL: Complex Analysis –https://nptel.ac.in
2. NPTEL: Probability and Statistics –https://nptel.ac.in
3. MIT OpenCourseWare – Complex Variables with Applications:https://ocw.mit.edu
4. MIT OpenCourseWare – Probability and Statistics:https://ocw.mit.edu
5. Python SymPy Documentation:https://docs.sympy.org
6. Google Colab for Python:https://colab.research.google.com
7. SciPy Documentation:https://docs.scipy.org
8. NumPy Documentation:https://numpy.org/doc

Recommended Free & Open-Source AI/Computational Tools:

1. Python (NumPy, SciPy, Matplotlib) – FREE on Google Colab for complex function visualization, numerical integration, and probability computations.
2. SymPy (free, open-source) – For symbolic complex analysis, series expansions, residue computation, and probability expressions.
3. scikit-learn (free, open-source) – For regression analysis, correlation computation, and exploratory data modelling.
4. Jupyter Notebook (free, open-source) – For interactive computation and lab documentation.
5. pandas (free, open-source) – For data manipulation and statistical analysis in probability and regression tasks.
6. GNU Octave (free, open-source) – MATLAB-like numerical computing for matrix and complex analysis workflows.
7. WolframAlpha (free tier) – For on-demand symbolic verification of integrals, series, and probability computations.

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26HMHS205T	HM	**Managerial Economics and Financial Analysis** (Common to all branches of Engineering)	2	0	0	0	2

Pre-Requisites: Basic mathematics and engineering economics concepts.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the fundamental concepts of demand, supply, market structures, and time value of money to solve basic engineering economic decision-making problems. (L3)

CO2: Apply economic theories of demand forecasting, cost analysis, and production functions to engineering project decision-making scenarios. (L3)

CO3: Analyse financial statements, cost-volume-profit relationships, and capital budgeting decisions using traditional and AI-powered financial tools. (L4)

CO4: Evaluate investment alternatives using NPV, IRR, and Payback Period methods and assess business viability under uncertainty using scenario analysis. (L5)

CO5: Create a comprehensive mini-project business plan incorporating market analysis, cost estimation, financial projections, and AI-assisted forecasting. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Introduction to Managerial Economics:

Managerial Economics – Meaning, Nature, Scope and significance for Engineers – Economic Analysis - Micro vs. Macroeconomics; positive vs. normative economics - Demand Analysis – Concept, Determinants, Demand Function, Law of Demand - Elasticity of Demand – Types, Measurement, Business Applications; Demand Forecasting – Meaning, Need, Methods.

AI Integration: AI Tool: ChatGPT – generate demand scenarios for an engineering product (e.g., solar panels) and Interpret price elasticity, AI Tool: Google Sheets + FORECAST function – implement Moving Average and Exponential Smoothing for a sample sales dataset, Activity: Use publicly available CMIE or RBI data; apply AI-assisted regression in Google Colab to forecast demand

Unit II: Production and Cost Analysis:
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: AI Tool: ChatGPT – generate a cost function for a manufacturing scenario and identify optimal Production level, Activity: Collect cost data of a local small enterprise; perform AI-assisted CVP analysis and present findings, Data Tool: Plotly (open-source Python) or Google Data Studio – visualise cost curves

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

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

Unit V: Financial Accounting and Analysis:
Principles and Digital Accounting Systems; Journal, Ledger, Trial Balance; Final Accounts – Trading Account, Profit and Loss Account, Balance Sheet with Simple Adjustments; Financial Analysis – Liquidity Ratios, Activity Ratios, Capital Structure Ratios, Profitability Ratios.
AI Integration: Automated financial statement preparation using Python (pandas), AI Tool: ChatGPT – interpret a company's financial ratios from a pasted balance sheet and income Statement, Machine learning for financial ratio prediction and company health assessment, Project: Download annual report of an Indian engineering company (BHEL, L&T, Infosys) and Perform ratio analysis.

Text Books:

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

Reference Books:

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

Online Resources

1. https://nptel.ac.in/courses/110105063
2. https://nptel.ac.in/courses/110107114
3. https://www.coursera.org/learn/economics
4. https://www.edx.org/course/financial-accounting
5. https://www.khanacademy.org/economics-finance-domain
6. https://swayam.gov.in

Recommended Free & Open-Source AI/Computational Tools:

1. ChatGPT + Statista- Use AI tools to forecast demand for a consumer product using historical data
2. Google Gemini- AI-generated adaptive quiz on types and measurement of elasticity
3. Microsoft Copilot- Build a Break-Even model using Excel + Copilot for sensitivity analysis
4. Wolfram Alpha / Tableau- Plot and interpret cost curves using AI-assisted data tools
5. Claude AI / ChatGPT- Generate real-world case studies of oligopoly (e.g., Telecom sector) using AI
6. Google Gemini- Evaluate a digital start-up's business model using AI-generated insights
7. Co-pilot + Wolfram Alpha- Use AI to compare NPV, IRR for two competing engineering projects
8. Microsoft Copilot- Explore how changes in cash flows affect IRR using AI-powered spreadsheets
9. Tally AI / Zoho AI- Prepare journal, ledger, and trial balance using Tally AI / Zoho Books
10. Tableau AI- Construct a financial ratio dashboard from a company's annual report using Tableau AI

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26ES01201T	ES	Sustainability And Green Engineering (AI Integrated) (Common to all branches of Engineering)	2	0	0	0	2

Pre-Requisites: Basic knowledge of Chemistry, Environmental Science, Physics, and Mathematics, along with an understanding of sustainability concepts and engineering applications.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the principles of sustainability, SDGs, carbon cycle, and greenhouse gas accounting to assess the environmental impacts of civil engineering materials and activities, with focus on the cement and steel industries. (L3)

CO2: Apply knowledge of sustainable construction materials including supplementary cementitious materials, recycled aggregates, geopolymer concrete, and alternative binders to evaluate their environmental performance using life cycle assessment methods. (L3)

CO3: Analyze the embodied energy and operational energy of building systems, perform energy calculations for building components, and compare the environmental performance of design alternatives using computational tools. (L4)

CO4: Evaluate green building rating systems (LEED, GRIHA, ECBC), assess compliance of building designs with energy codes, and compare the sustainability performance of different building design strategies using simulation and AI tools. (L5)

CO5: Design sustainable civil engineering solutions integrating circular economy principles, C&D waste management, green infrastructure, and AI-driven optimization tools for minimizing carbon footprint and maximizing resource efficiency.(L6)

CO – PO Articulation Matrix:

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

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

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 fiber-reinforced composites as sustainable alternatives. 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). Energy concept: primary energy, renewable vs. non-renewable energy. 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. construction
Free/Open-Source AI Tools: EnergyPlus (free, US DOE simulation tool); OpenStudio (free); Python (scikit-learn, NumPy) – Google Colab; openLCA (free)

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 (Leadership in Energy and Environmental Design): credit categories, certification levels; GRIHA (Green Rating for Integrated Habitat Assessment): India's national system, parameters; IGBC rating systems; BEE star rating for commercial buildings. Passive design strategies: building orientation, natural ventilation, day lighting, 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 sensor for real-time energy monitoring. Case Study: AI-optimized green building design

Free/Open-Source AI Tools: Energy Plus (free); Open Studio (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 utilization. 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)

Text Books:

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. Langston C.A. and Ding G.K.C., Sustainable Practices in the Built Environment, Butterworth Heinemann, 2011.
2. Spence W.P., Construction Materials, Methods and Techniques, Yesdee Publication, 3rd Edition, 2012.
3. ECSBC 2024 (Energy Conservation and Sustainability Building Code), Bureau of Energy Efficiency, Government of India.
4. National Building Construction Standards,2026, Bureau of Indian Standards.
5. Goswami D., Kreith F. and Kreider J., Principles of Solar Engineering, Taylor & Francis, 2000.

Online Resources:

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

Recommended Free & Open-Source AI/Computational Tools:

1. openLCA (free, open-source) – Life Cycle Assessment software for embodied carbon calculation
2. EnergyPlus (free, US DOE) – Building energy simulation for operational energy analysis
3. OpenStudio (free, NREL) – Building energy modeling interface for EnergyPlus
4. Python (scikit-learn, Pandas, NumPy) on Google Colab – ML for energy prediction and material selection
5. QGIS (free) – Green infrastructure planning and spatial sustainability analysis
6. i-Tree Tools (free, USDA) – Urban tree carbon sequestration quantification
7. SimaPro (student trial) – Life cycle assessment platform

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26EC02201T	EC	Basic Electrical & Electronics Engineering	3	0	0	0	3

Pre-requisite: Basic knowledge of mathematics , physics, and fundamental electrical concepts.

PART A: BASIC ELECTRICAL ENGINEERING

Course Objectives:

- To introduce the fundamental concepts and principles of DC and AC electrical circuits, electrical machines, and measuring instruments.
- To familiarize students with energy resources, power generation systems, electricity consumption and billing, and electrical safety practices.
- To develop the ability to analyze and evaluate electrical systems using appropriate simulation, measurement, and AI-assisted tools for circuit analysis, machine diagnostics, energy analytics, and smart monitoring.

Course Outcomes (COs):

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

CO1: Analyse DC and AC electrical circuits using fundamental circuit laws, phasor concepts, power relationships using AI-assisted simulation and analysis tools. (L4)

CO2: Analyze the construction, operating principles, and applications of electrical machines and measuring instruments using appropriate analytical, simulation, and AI-assisted diagnostic techniques. (L4)

CO3: Evaluate different energy-generation systems, domestic electricity consumption and billing, and electrical safety practices using AI-assisted energy analytics and smart monitoring tools. (L4)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	3	2	1				1	1	1
CO2	3	3	2	2	3	2	1				1	1	1
CO3	3	3	3	2	3	2	1				1	1	1

Unit I: DC&AC Circuits:

DC Circuits: Electrical circuit elements (R, L and C), Ohm's Law and its limitations, KCL & KVL, series, parallel, series-parallel circuits, Super Position theorem, Simple Numerical problems. **AC Circuits:** A.C. Fundamentals: Equation of AC Voltage and current, waveform, time period, frequency, amplitude, phase, phase difference, average value, RMS value, form factor, peak factor, Voltage and current relationship with phasordiagrams in R, L, and C circuits, Concept of Impedance, Analysis of R-L, R-C, R-L-C Series circuits, Active power, reactive power and apparent power, Concept of power factor (Simple Numerical problems).

AI Integration: Introduction to AI-assisted Electrical Circuit Analysis, Intelligent Waveform and Phasor Diagram Analysis, Machine Learning Applications for Voltage and Current Prediction, AI-based Power Factor Monitoring and Correction, AI-enabled Circuit Simulation Tools and Applications

Suggested AI Tools: MATLAB, PSpice, LTspice, Python

Unit II: Machines and Measuring Instruments:

Machines: Construction, principle and operation of (i) DC Motor, (ii) DC Generator, (iii) Single Phase Transformer, (iv) Three Phase Induction Motor and (v) Alternator, Applications of electrical machines.

Measuring Instruments: Construction and working principle of Permanent Magnet Moving Coil (PMMC), Moving Iron (MI) Instruments and Wheat Stone Bridge.

AI Integration: Introduction to AI in Electrical Machines, AI-based Predictive Maintenance of Motors and Transformers, Smart Sensor-based Monitoring of Electrical Parameters, AI-based Error Detection in Measuring Instruments, Intelligent Digital Measuring Instruments

Suggested AI Tools: MATLAB Simulink, TensorFlow, LabVIEW,Scikit-learn

Unit III: Energy Resources, Electricity Bill &Safety Measures:

Energy Resources: Conventional and non-conventional energy resources; Layout and operation of various Power Generation systems: Hydel, Nuclear, Solar & Wind power generation.

Electricity Bill: Power rating of household appliances including air conditioners, PCs, Laptops, Printers, etc. Definition of "unit" used for consumption of electrical energy, two-part electricity tariff, calculation of electricity bill for domestic consumers.

Equipment Safety Measures: Working principle of Fuse and Miniature Circuit Breaker (MCB), merits and demerits. Personal safety measures: Electric Shock, Earthing and its types, Safety Precautions to avoid shock.

AI Integration: Introduction to Smart Energy Systems and AI Applications, Smart Metering and Intelligent Electricity Consumption Monitoring, AI-assisted Electricity Bill Prediction and Energy Analytics, Intelligent Energy Management Systems for Smart Homes, Smart Electrical Safety Monitoring Systems

Suggested AI Tools: Power BI, HOMER Pro, Arduino IDE, Google Colab

Text Books:

1. Basic Electrical Engineering, D. C. Kulshreshtha, TataMcGrawHill,2019, First Edition
2. Power System Engineering, P. V. Gupta, M. L. Soni, U. S. Bhatnagar and A. Chakrabarti, Dhanpat Rai & Co, 2013
3. Fundamentals of Electrical Engineering, Rajendra Prasad, PHI publishers,2014,Third Edition

Reference Books:

1. Basic Electrical Engineering, D. P. Kothariand I.J.Nagrath, McGrawHill, 2019, Fourth Edition
2. Principles of Power Systems,V. K. Mehtha, S. Chand Technical Publishers,2020
3. Basic Electrical Engineering, T. K. Nagsarkar and M. S. Sukhija, Oxford University Press, 2017
4. Basic Electrical and Electronics Engineering, S. K. Bhatacharya, Person Publications, 2018, Second Edition.

Online Resources:

1. https://nptel.ac.in/courses/108105053
2. https://nptel.ac.in/courses/108108076

PART A: BASIC ELECTRONIC ENGINEERING

Pre-requisite: Basic knowledge of mathematics, physics, electrical circuits, and fundamental concepts of electronic d

Course Objectives:

- To understand semiconductor devices like PN junction diode, Zener diode, BJT and analyse their characteristics and applications in basic electronic circuits.
- To apply semiconductor device knowledge to design and analyse rectifier circuits, amplifiers, and instrumentation systems.
- To understand digital electronics fundamentals like number systems, logic gates, Boolean algebra, and sequential circuits.

Course Outcomes (COs):

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

CO1: Analyse the characteristics of PN junction diode, Zener diode, and BJT configurations to evaluate their behaviour and select appropriate devices for basic electronic applications. (L4)

CO2: Apply knowledge of rectifiers, filters, amplifiers, and instrumentation systems to explain circuit operation and simulate waveform behaviour using Python tools. (L3)

CO3: Design basic combinational and sequential digital circuits using logic gates, Boolean algebra, and flip-flops, and verify logic behaviour using LLM and Python simulation tools. (L3)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	2	3	2	1				1	1	1
CO2	3	3	2	2	3	2	1				1	1	1
CO3	3	3	3	2	3	2	1				1	1	1

Unit I: Semiconductor Devices:

Introduction — Evolution of electronics: vacuum tubes to nano electronics. Characteristics of PN Junction Diode. Zener Effect — Zener diode and its characteristics. Bipolar Junction Transistor — CB, CE, CC configurations and characteristics. Elementary treatment of small signal CE amplifier.

Suggested Tasks:

- Use AI/LLM tools to explain the construction of a PN junction diode, and the Zener effect; ask it to describe the V-I characteristics of a PN junction and Zener diode with sketches, and compare their operating regions in a short table.
- Use a provided Python template in Google Colab to plot the ideal V-I characteristic curve of a PN junction diode using the diode equation; vary the saturation current and thermal voltage and observe how the curve shape changes, then superimpose the Zener diode characteristic to compare forward and reverse breakdown behaviour.
- Use AI/LLM tool to explain CB, CE, and CC BJT configurations — input/output characteristics, current gain, and typical applications; prepare a comparison table covering input impedance, output impedance, current gain, voltage gain, and phase relationship, and ask the LLM to describe the small signal CE amplifier operation in simple terms.

Unit II: Basic Electronic Circuits and Instrumentation:

Block diagram of a DC power supply, working of a full wave bridge rectifier, capacitor filter (no analysis), working of simple Zener voltage regulator. Amplifiers: block diagram of Public Address system, circuit diagram and working of common emitter (RC coupled) amplifier with frequency response, concept of voltage divider biasing. Electronic Instrumentation: block diagram of an electronic instrumentation system.

Suggested Tasks:

- Use AI/LLM tool to explain the block diagram of a DC power supply and the working of a full wave bridge rectifier; ask it to describe the role of a capacitor filter and a Zener voltage regulator in simple terms, and prepare a labelled block diagram description with the function of each stage (transformer, rectifier, filter, regulator).
- Use a provided Python template in Google Colab to simulate a full wave bridge rectifier output: generate a sinusoidal input signal, apply ideal diode rectification, and plot the input and rectified output waveforms; then add a smoothing capacitor and observe how the ripple voltage changes with different capacitor values.
- Use AI/LLM tool to explain the working of a common emitter RC coupled amplifier — the role of coupling capacitors, bypass capacitor, and voltage divider bias network; ask it to describe the frequency response (low, mid, and high frequency regions) and sketch the gain-vs-frequency curve, then prepare a short note on the block diagram of an electronic instrumentation system.

Unit III: Digital Electronics:

Over view of number systems. Logic gates including universal gates. BCD codes, Excess-3 code, Gray code, Hamming code. Boolean algebra — basic theorems and properties. Truth tables and functionality of logic gates: NOT, OR, AND, NOR, NAND, XOR, and XNOR. Simple combinational circuits: half adder and full adder. Introduction to sequential circuits — flip-flops, registers, and counters (elementary treatment only).

Suggested Tasks:

- Use an AI/LLM toolto explain the binary, octal, and hexadecimal number systems and conversions between them; ask it to describe BCD, Excess-3, Gray, and Hamming codes with examples, and solve

2–3 numerical problems on code conversion (e.g., decimal to BCD, binary to Gray code); verify the results manually.

- Use a provided Python template in Google Colab to generate truth tables for all seven logic gates (NOT, OR, AND, NOR, NAND, XOR, XNOR) for 2-variable and 3-variable inputs; verify De Morgan's theorems programmatically by comparing both sides of the identities, and implement a half adder and full adder using basic gate functions.
- Use an AI/LLM toolto explain SR, D, JK, and T flip-flops in simple terms — truth tables, excitation tables, and the concept of clock-triggered operation; ask it to describe the working of a 4-bit shift register and a 3-bit ripple counter with state sequences, and prepare a concise comparison table of flip-flop types.

Instructions:

1. The suggested tasks can be modified based on the decision of the subject expert.
2. The suggested tasks can be part of assignments using appropriate AI/LLM tools and Python/MATLAB-based simulations, with the choice of platforms and machine learning models decided by the subject expert to enhance conceptual understanding and practical AI-enabled engineering skills.

Text Books:

1. Boylestad, R.L. and Nashelsky, L., "Electronic Devices and Circuit Theory", 11th Edition, Pearson Education, 2013.
2. Floyd, T.L., "Digital Fundamentals", 11th Edition, Pearson Education, 2015.

Reference Books:

1. Sedra, A.S. and Smith, K.C., "Microelectronic Circuits", 7th Edition, Oxford University Press, 2014.
2. Millman, J. and Halkias, C.C., "Electronics Devices and Circuits", Tata McGraw-Hill, 2008.
3. Morris Mano, M. and Ciletti, M.D., "Digital Design", 5th Edition, Pearson Education, 2013.
4. Razavi, B., "Fundamentals of Microelectronics", 2nd Edition, Wiley, 2013.

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01203T	**PC**	**Surveying (AI Integrated)**	**2**	**0**	**2**	**0**	**3**

Pre-Requisites: Basics of Engineering Graphics, Engineering Mathematics and Engineering Mechanics

Course Objectives:

- To explain the principles, classification, and methods of surveying for measuring horizontal distances, angles, and vertical levels.
- To apply methods of chain surveying, compass surveying, levelling, and theodolite surveying to field problems.
- To use modern surveying instruments: Total Station, GPS/GNSS, and drone survey systems with understanding of their working principles.
- To apply photogrammetry and LiDAR principles for large-scale mapping and volumetric analysis.
- To integrate AI and machine learning tools for automated data processing, error correction, and feature extraction in surveying workflows.

Course Outcomes (COs):

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

CO1: Apply surveying principles, classification, and methods to select appropriate instruments and perform basic chain, compass, and plane table survey tasks. (L3)

CO2: Apply levelling, contouring, theodolite traversing, and curve-setting methods to solve surveying problems, and compute areas and volumes using standard surveying data. (L3)

CO3: Analyse field survey data using computational tools to perform traverse adjustments, error analysis, and generate contour maps and digital elevation models. (L4)

CO4: Evaluate modern surveying technologies including Total Station, GPS/GNSS, LiDAR, and UAV-based photogrammetry systems, and compare their relative accuracy, efficiency, and applicability for various civil engineering survey applications. (L5)

CO5: Design survey plans and create automated processing pipelines for geospatial data using AI/ML tools including deep learning for feature extraction, point cloud processing, and change detection in remote sensing imagery. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Introduction, Chain & Compass Surveying:

Core Content: Introduction to surveying: objectives, classification, principles (working from whole to part, consistency of measurements), sources of error. Chain surveying: types of chains and tapes, ranging, offsets, obstacles. Measurement corrections: tape corrections for temperature, pull, sag, slope. Compass surveying: prismatic compass, bearings (WCB and QB systems), local attraction, magnetic declination and dip, traversing with compass, Bowditch's rule. Plane table surveying: radiation, intersection, traversing, resection (Bessel’s method).

AI Integration: Introduction to digital data collection: electronic field books, cloud-based survey platforms. Data quality assessment using Python: statistical outlier detection in survey measurements. Introduction to AI in surveying: overview of how ML is transforming geospatial data collection. Computer-aided traverse computations using Python/spreadsheets.

Free/Open-Source AI Tools: QGIS (free, open-source GIS platform); Python (NumPy, Pandas) on Google Colab; Trimble Sketch (free); Libre Office Calc for computations.

Unit II: Levelling, Contouring, Areas & Volumes:

Core Content: Levelling: types of levels (dumpy, auto level), staff, booking methods (HI and rise & fall), fly levelling, profile levelling, reciprocal levelling, effect of curvature and refraction. Contouring: characteristics, methods (direct and indirect), interpolation, uses of contour maps. Areas: mid-ordinate rule, average-ordinate rule, Simpson's rule, coordinates method, planimeter. Volumes: prismoidal formula, end-area method, capacity of reservoirs, mass haul curve.

AI Integration: Automated levelling data processing using Python. Contour map generation using Python (Matplotlib, SciPy interpolation). DEM (Digital Elevation Model) creation from levelling data. Introduction to AI-based contouring: how point clouds are processed using ML. Machine learning for automatic volume computation from survey data. regression coefficients and correlation strength.

Free / Open-Source AI Tools: QGIS with contour plugin (free); Python (SciPy, Matplotlib) – Google Colab; Global Mapper (student); MicroSurveyFieldGenius (free trial)

Unit III: Theodolite, Traversing & Curves:

Core Content: Theodolite: construction, temporary adjustments, measurement of horizontal angles (repetition and reiteration), vertical angles, stadia tacheometry. Traversing: methods, traverse computations (Bowditch's rule, transit rule), Gale's table, omitted measurements. Trigonometric levelling: accessible and inaccessible bases. Types of curves: simple, compound, reverse; elements; setting out: ordinates from long chord, Rankine's deflection angle method. Introduction to route surveys.

AI Integration: Automated traverse computation and error distribution using Python. AI-based identification of blunders in survey measurements using statistical hypothesis testing. Introduction to machine learning for tacheometric data processing. Python-based curve setting-out calculations. Digital theodolite data integration with GIS.

Free/Open-Source AI Tools: Trimble Business Centre (free student version); Python (SciPy, NumPy); QGIS; online traverse calculators (free)

Unit IV: Modern Surveying Methods – Total Station, GPS & Drone Survey:

Core Content: EDM (Electronic Distance Measurement): principles, types. Total Station: components, functions, data storage, downloading. Applications: topographic surveys, setting out. GPS/GNSS: principles of satellite ranging, GPS constellation, signal structure, DGPS, RTK-GPS, accuracy levels, applications. Introduction to Drone (UAV) survey: types of UAVs, photogrammetric mission planning, ground control points (GCPs), flight planning software. LiDAR: principles, types (airborne and terrestrial), point cloud data, data products.

AI Integration: AI-enhanced GNSS processing: ML for multipath error correction. Automated flight path planning for drones using AI algorithms. Introduction to point cloud processing: PCL (Point Cloud Library, free). Deep learning for 3D reconstruction from drone images using Open Drone Map (free). AI for GCP-free photogrammetry using deep learning feature matching.

Free/Open-Source AI Tools: OpenDroneMap (free, open-source drone mapping software); QGIS (free); AgisoftMetashape (student version/free educational); WebODM (free, open-source cloud photogrammetry)

Unit V: Photogrammetry, Remote Sensing & AI in Surveying:

Core Content: Photogrammetry: basic concepts, types (terrestrial, aerial), perspective geometry of aerial photographs, relief and tilt displacements. Flight planning: overlap, scale, camera focal length. Stereoscopy: parallax equations, height determination. Radial triangulation. Mosaics and map substitutes. Introduction to remote sensing: electromagnetic spectrum, sensors, satellite platforms. Applications: land use mapping, change detection, disaster management.

AI Integration: Structure from Motion (SfM): AI-based 3D reconstruction from overlapping photographs. Deep learning for automatic feature extraction: roads, buildings, vegetation classification from satellite imagery. Google Earth Engine (free, cloud-based) for large-scale satellite data processing. ML-based change detection for land use monitoring. AI-assisted quality control in survey data: automated blunder detection, coordinate transformation using neural networks. Point cloud classification using ML: ground, vegetation, building separation.

Free/Open-Source AI Tools: Google Earth Engine (free cloud platform); QGIS (free, open-source GIS); OpenDroneMap/WebODM (free photogrammetry); CloudCompare (free point cloud processing); Python (rasterio, geopandas, OpenCV) on Google Colab

Text Books:

1. Duggal S.K., Surveying Vol. 1 & 2, Tata McGraw Hill Publishing Co. Ltd., 5th Edition, 2019.
2. Venkatramaiah C., Textbook of Surveying, Universities Press, 1st Edition, 2011.

Reference Books:

1. Punmia B.C., Jain A.K. and Jain A.K., Surveying Vol. 1, 2 & 3, Laxmi Publications, 18th Edition, 2024.
2. Chandra A.M., Plane Surveying and Higher Surveying, New Age International, 3rd Edition, 2015.
3. Basak N., Surveying and Levelling, Tata McGraw Hill, 4th Edition, 2014.
4. Wolf P.R. and Dewitt B.A., Elements of Photogrammetry with Applications in GIS, McGraw Hill, 3rd Edition, 2000.
5. Jensen J.R., Introductory Digital Image Processing, Pearson, 4th Edition, 2015.

Online Resources:

1. NPTEL: Surveying – https://nptel.ac.in/courses/105107123
2. NPTEL: Geomatics Engineering – https://nptel.ac.in/courses/105105202
3. OpenDroneMap (free drone mapping): https://www.opendronemap.org
4. Google Earth Engine (free cloud GIS): https://earthengine.google.com
5. QGIS Official Documentation (free GIS): https://docs.qgis.org
6. CloudCompare (free point cloud): https://cloudcompare.org
7. NPTEL: Remote Sensing – https://archive.nptel.ac.in/courses/105/105/105105202

Recommended Free & Open-Source AI/Computational Tools:

1. QGIS 3.x – Free, open-source GIS and spatial analysis platform (qgis.org)
2. OpenDroneMap / WebODM – Free, open-source photogrammetry from drone images (opendronemap.org)
3. Google Earth Engine – Free cloud computing platform for geospatial analysis
4. CloudCompare – Free, open-source 3D point cloud processing (cloudcompare.org)
5. Python (geopandas, rasterio, OpenCV, shapely) – Google Colab (free)
6. AgisoftMetashape – Student/educational edition available; or free Meshroom (AliceVision)
7. Python scikit-learn – Machine learning for survey data classification and outlier detection

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01204T	PC	Strength of Materials (AI Integrated)	2	0	2	0	3

Pre-Requisites: Fundamentals of Engineering Mechanics and Engineering Mathematics

Course Objectives:

- To draw and interpret shear force and bending moment diagrams for various beam types and loading conditions.
- To calculate bending stresses, shear stresses, and torsional stresses in structural members.
- To analyse beam deflections using double integration, Macaulay's method, and moment area methods.
- To apply AI/ML tools for material property prediction, stress analysis, and structural health monitoring applications.

Course Outcomes (COs):

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

CO1: Apply the principles of shear force and bending moment to draw accurate SFD and BMD for cantilever, simply supported, and overhanging beams subjected to various load combinations, and solve flexural and shear stress problems for standard cross-sections. (L3)

CO2: Analyse the moments of Fixed and continuous beams (L4)

CO3: Analyze the deflection of beams using double integration, Macaulay's, and moment area methods, and evaluate the structural adequacy of beams with respect to allowable deflection limits. (L4)

CO4: Evaluate the buckling behaviour of columns using Euler's and Rankine-Gordon theories, assess stress states in thin and thick cylindrical pressure vessels using Lame's equations, and compare analytical results with computational/ML-based predictions. (L5)

CO5: Design computational solutions for structural analysis and SHM using Python and AI/ML tools, including regression models for property prediction, CNN-based crack detection, and data-driven digital twin frameworks for structural condition assessment. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Shear Force & Bending Moment Diagrams:

Core Content: Definition and types of beams and loads; concept of shear force and bending moment; sign conventions; point of contra-flexure; relation between load intensity, shear force, and bending moment (Schwedler's theorem). SFD and BMD for: cantilever beams, simply supported beams, overhanging beams – under point loads, UDL, UVL, partial UDL, couples, and combinations. Introduction to continuous beams: concept only.

AI Integration: Python-based automated SFD and BMD generation for any beam configuration (symbolic computation using SymPy). AI-assisted structural analysis: overview of how commercial FEM software automates SFD/BMD. Introduction to graph neural networks (GNNs) for structural element analysis (conceptual). Using OpenSees Python interface for beam analysis.

Free/Open-Source AI Tools: Python (SymPy, NumPy, Matplotlib) – Google Colab; OpenSees (free); online SFD/BMD calculators for verification; BeamNG (free structural visualizer)

Unit II: Fixed Beams & Continuous Beams :

Core Content: Fixed beams: fixed end moments (FEM) for standard loads – UDL, central point load, eccentric point load, UVL, couples. SFD and BMD for fixed beams. Deflection of fixed beams. Effect of sinking of support: analysis and SFD/BMD. Effect of rotation of a support. Three-moment equation (Clapeyron's theorem): derivation, application to two-span and multi-span continuous beams with and without settlement of supports.

AI Integration: Python SymPy for symbolic derivation of fixed end moments. Automated SFD/BMD generation for continuous beams using Python. Comparison of results with OpenSees model. ML for continuous beam moment prediction: training a regression model on a dataset of beam configurations. Introduction to influence lines for continuous beams using Python.

Free/Open-Source AI Tools: Python (SymPy, NumPy, Matplotlib) – Google Colab; OpenSees (free); SAP2000 student version; BeamNG structural analysis online (free)

Unit III: Flexural Stresses, Shear Stresses & Torsion:

Core Content: Theory of simple bending: assumptions, derivation of flexure formula ($M/I = \sigma/y = E/R$). Neutral axis, section modulus. Bending stress distribution in rectangular, circular, I, T, angle, and channel sections. Design of simple beams based on bending. Shear stress formula derivation. Shear stress distribution across rectangular, circular, I-section, T-section. Torsion of circular shafts (solid and hollow): derivation of torsion formula, power transmission.

AI Integration: AI for cross-section optimization: introduction to topology optimization concepts using Python. ML-based prediction of bending moment capacity from cross-section dimensions. Computer vision for detecting flexural cracks in beams: introduction to image classification with Tens or Flow (free). Using Python to automate section modulus calculations for standard steel sections.

Free/Open-Source AI Tools: Python (scikit-learn, TensorFlow/Keras) on Google Colab; FreeCAD (free, for cross-section visualization); AutoCAD student version; SkyCiv free structural analysis online tool.

Unit IV: Deflection of Beams:

Core Content: Introduction to beam deflection: importance in structural design. Double integration method: derivation of differential equation of elastic curve, slopes and deflections for cantilevers and simply supported beams under standard loads. Macaulay's method for beams with discontinuous loading. Moment area method: Mohr's theorems, application to cantilevers and propped cantilevers. Conjugate beam method (introduction).

AI Integration: ML for beam deflection prediction: regression model to predict maximum deflection from beam parameters (L, EI, loads). Python simulation of deflected shape of beams using numerical methods (Euler integration). Introduction to neural networks for real-time structural deflection monitoring (digital twin concept). Experimental vs. theoretical comparison using data from strain gauges: Python data analysis.

Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib) – Google Colab; SkyCiv free beam calculator for verification; OpenSees; Excel/LibreOffice Calc for Macaulay's method tables.

Unit V: Columns, Cylinders & AI-Based Structural Health Monitoring:

Core Content: Columns: types of columns; Euler's theory: critical load for various end conditions, slenderness ratio, Euler's critical stress, limitations; Rankine-Gordon formula; eccentric loading, secant formula. Thin cylinders: hoop and longitudinal stresses, volumetric strain. Thick cylinders: Lame's equations, distribution of hoop and radial stresses, compound cylinders.

AI Integration: AI in Structural Health Monitoring (SHM): vibration-based SHM using accelerometer data and ML (anomaly detection with scikit-learn). Computer vision for crack detection in concrete: introduction to convolutional neural networks (CNNs) for crack image classification using pre-trained models (TensorFlow). IoT sensors and real-time data analytics for structural condition assessment. Digital twins for bridges: conceptual introduction. Case study: SHM of a column using Python.
Free/Open-Source AI Tools: Python (scikit-learn, TensorFlow, OpenCV) – Google Colab (free); TensorFlow Hub (pre-trained models for crack detection) – Google Colab; OpenSees (free structural analysis framework with Python interface); SMRF: Structural Monitoring and Response Framework (open-source)

Text Books:

1. Bansal R.K., Strength of Materials, Laxmi Publications, 16th Edition, 2022.
2. Basavarajaiah B.S. and Mahadevappa P., Strength of Materials, Universities Press, 3rd Edition, 2010.
3. Punmia B.C., Strength of Materials and Mechanics of Structures, Khanna Publications, New Delhi.

Reference Books:

1. Rajput R.K., Textbook of Strength of Materials (Mechanics of Solids), S. Chand & Co., 7th Edition, 2022.
2. Popov E.P., Mechanics of Solids, Prentice Hall, 2nd Edition, 2015.
3. Beer F.P. and Johnston E.R., Mechanics of Materials, McGraw Hill India, 8th Edition, 2020.
4. Timoshenko S. and Gere J., Mechanics of Materials, CBS Publishers, 2006.
5. Srinath L.S., Advanced Mechanics of Solids, McGraw Hill Education, 3rd Edition, 2017.

Online Resources:

1. NPTEL: Strength of Materials – https://nptel.ac.in/courses/112105147
2. MIT OpenCourseWare – Mechanics of Materials: https://ocw.mit.edu/courses/2-002
3. SkyCiv Free Structural Analysis: https://skyciv.com/structural-analysis-software
4. Python SymPy for Symbolic Beam Analysis: https://docs.sympy.org
5. Google Colab for Python: https://colab.research.google.com
6. OpenSees Official Site: https://opensees.berkeley.edu
7. Crack Detection ML Tutorial (Google Colab/TensorFlow): https://www.tensorflow.org/tutorials

Recommended Free & Open-Source AI/Computational Tools:

1. Python (SymPy, NumPy, SciPy, Matplotlib) – FREE on Google Colab for beam analysis automation
2. TensorFlow / Keras (free, open-source) – For crack detection CNN on Google Colab
3. scikit-learn (free, open-source) – For structural property regression and SHM anomaly detection
4. OpenSees (free, open-source) – Structural analysis with Python interface
5. FEniCS (free, open-source) – FEM-based stress analysis
6. SkyCiv Free Beam Calculator – Online verification tool (free tier available)
7. OpenCV (free, open-source) – For image processing in visual crack inspection

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26EC02201P	EC	Basic Electrical & Electronics Engineering Lab	0	0	0	2	1

Pre-Requisites: Basic knowledge of mathematics, physics, electrical circuits, and fundamental concepts of electronic devices.

Part-A: Electrical Engineering

- To develop practical skills in electrical circuit analysis, measurements, electrical machines, and power measurement through laboratory experiments.
- To familiarize students with electrical safety practices, domestic energy calculations, and hardware/software tools for monitoring and analyzing electrical systems.
- To enhance analytical, experimental, and problem-solving skills using AI-assisted and open-source simulation tools for electrical analysis and intelligent data interpretation.

Course Outcomes (COs):

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

CO1: Understand and apply fundamental electrical circuit concepts, measurement techniques, wiring practices, and operating principles of electrical machines and transformers. (L3)

CO2: Apply theoretical concepts and mathematical methods to measure and evaluate electrical parameters such as resistance, power, and power factor using appropriate instruments. (L3)

CO3: Analyse and design electrical circuits, machines, measuring instruments, and wiring systems based on their characteristics and practical requirements. (L4)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	2	2	3	1	3	3	3		1	1	1
CO2	3	3	2	3	3	1	3	3	3		1	1	1
CO3	3	3	3	3	3	1	3	3	3		2	1	1

List of Experiments:

1. Verification of KCL and KVL
2. Verification of Superposition theorem
3. Measurement of Resistance using Wheatstone bridge
4. Magnetization Characteristics of DC shunt Generator
5. Measurement of Power and Power factor using Single-phase wattmeter
6. Measurement of Earth Resistance using Megger
7. Calculation of Electrical Energy for Domestic Premises

AI Integrated Topics:

- AI-assisted circuit analysis and verification
- Intelligent prediction of circuit responses using AI tools
- AI-based resistance estimation and error analysis
- AI-assisted prediction of magnetization characteristics
- AI-based power factor monitoring and analysis
- Intelligent earthing fault detection using AI techniques
- AI-assisted domestic energy consumption and electricity bill prediction

Suggested AI Tools:

- LTspice, Scilab, KiCad + Ngspice and Python

Reference Books:

1. Basic Electrical Engineering, D.C. Kulshreshtha, Tata McGraw Hill,2019, First Edition
2. Power System Engineering, P.V. Gupta, M.L. Soni, U.S. Bhatnagar and A.Chakrabarti, Dhanpat Rai & Co, 2013
3. Fundamentals of Electrical Engineering, Rajendra Prasad PHI publishers 2014, Third Edition.

Part-B: Basic Electronics Engineering

Pre-requisites: Basic knowledge of mathematics, physics, electrical circuits, and fundamental concepts of electronic devices.

Course Objectives:

- To introduce V-I and transfer characteristics of diodes and BJT and extract device parameters.
- To develop understanding of rectifier circuits, biasing networks, and amplifier design.
- To familiarize students with LLM-based circuit interpretation and ML techniques for parameter estimation..

Course Outcomes (COs):

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

CO1: Characterize electronic devices by plotting V-I curves and extracting parameters. (L3)

CO2: Design and evaluate rectifier and amplifier circuits and measure key performance metrics. (L3)

CO3: Apply ML regression and classifiers to estimate device parameters and automate analysis. (L3)

CO – PO Articulation Matrix:

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

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	3	3	2	3	3	3			1	1
CO2	3	3	3	3	3	2	3	3	3			1	1
CO3	3	3	2	3	3	2	3	3	3			1	1

List of Experiments:

1. Plot V-I characteristics of PN Junction Diode A) Forward bias B) Reverse bias.
2. Plot V–I characteristics of Zener Diode and its application as voltage Regulator.
3. Implementation of half wave and full wave rectifiers
4. Plot Input & Output characteristics of BJT in CE and CB configurations
5. Frequency response of CE amplifier.
6. Simulation of RC coupled amplifier with the design supplied
7. Verification of Truth Table of AND, OR, NOT, NAND, NOR, Ex-OR, Ex-NOR gates using ICs.
8. Verification of Truth Tables of S-R,J-K &D flip flops using respective ICs

AI/LLM and/or Python / MATLAB Tasks:

Any three experiments need to be implemented using appropriate AI/LLM tools and/or Python/MATLABbased simulations.

1. a) Usean AI/LLM tool 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.

   b) Use ML/Python to Train a regression model to fit the diode equation (Shockley equation) to the measured V-I data and estimate saturation current (IS) and ideality factor (n).

2. a) Use an AI/LLM tool to explain the Zener breakdown mechanism, the difference between avalanche and Zener breakdown, and how the diode maintains voltage regulation under varying load.
   b) Use ML/Python to Build a regression model to predict regulation performance (output voltage vs. load current) from measured characteristic data.
3. a) Use an AI/LLM tool 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.
   b) Use Python/MATLAB simulation to predict ripple voltage and efficiency. Apply regression to model ripple factor as a function of load resistance and capacitance value.
4. a) Use an AI/LLM tool to explain CE and CB input/output characteristics, the concept of current gain β and α, the Early effect, and compare the two configurations for impedance and gain.
   b) Use ML/Python to Train a regression model on CE output characteristic data to estimate β and output resistance. Build a parameter prediction model for CB configuration to estimate α and input/output resistance.
5. a) Use an AI/LLM tool to analyse the BJT CE amplifier frequency response, explain low-frequency and high-frequency cutoff causes, and interpret the Bode plot.
   b) Use ML/Python to curve fitting (polynomial or spline regression) to model the frequency response and estimate the -3 dB bandwidth automatically from simulation or measured data.
6. a) Use an AI/LLM tool to explain RC coupling, the purpose of coupling and bypass capacitors, inter-stage loading effects, and how these shape the frequency response.
   b) Combine LLM-guided design with a regression/optimisation model to predict midband gain and bandwidth for different RC values, enabling design optimization
7. a) Use an AI/LLM tool to explain the Boolean algebra behind each gate, derive truth tables from first principles, and discuss NAND and NOR as universal gates with implementation examples.
   b) Use ML/Python to Train a simple classifier (decision tree or logic network) using the truth table data as training set, and verify that the classifier reproduces correct outputs for all input combinations.
8. a) Use an AI/LLM tool to explain the operation of SR, JK, and D flip-flops, derive their characteristic equations, and explain why JK flip-flop eliminates the invalid state of SR flip-flop.
   b) Use ML/Python to Train a sequence prediction model (e.g., simple RNN or state-table lookup) using flip-flop state transition data to predict Q(next) given current state Q and inputs; verify it matches theoretical characteristic equations.

Instructions:

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

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01203P	PC	Surveying Lab (AI Integrated)	0	0	0	3	1.5

Pre-Requisites:

Course Objectives:

- To familiarize students with basic linear and angular measuring instruments and their field use.
- To develop skills in conducting chain, compass, levelling, and theodolite field surveys.
- To train students in the use of Total Station for modern field surveying.
- To introduce drone-based survey data collection and processing workflows.
- To apply AI/computer-based tools for automated survey data processing and analysis.

Course Outcomes (COs):

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

CO1: Handle various linear, angular, and levelling instruments correctly and conduct basic chain, compass, and plane table surveys. (L3)

CO2: Measure distances, angles, and elevations accurately and apply appropriate booking/computation methods for field data reduction. (L3)

CO3: Operate a Total Station for coordinate measurement, area computation, and setting-out tasks, and interpret modern digital survey data. (L4)

CO4: Process surveying field data using computer-based tools (Python/QGIS) to generate contour maps, traverse plots, and digital survey plans. (L5)

CO5: Evaluate drone survey workflows and AI-based data processing tools, and demonstrate awareness of modern geospatial technologies for civil engineering applications. (L6)

CO – PO Articulation Matrix:

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

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

List of Field Work/Lab Experiments:

1. Chain Survey: Road profile with offsets (chain, tapes, ranging rods).
2. Compass Surveying: Determination of distance between two inaccessible points using prismatic compass and Bowditch's rule for traverse adjustment.
3. Plane Table Survey: Finding area of a given boundary using radiation method and two-point resection (Bessel's method).
4. Fly Levelling – HI Method: Differential levelling along a profile (minimum 5 stations).
5. Fly Levelling – Rise & Fall Method: Same data, comparison of both booking methods, closure error computation.
6. Theodolite Survey: Measuring horizontal angles by repetition method; measuring vertical angles.
7. Theodolite Survey: Finding distance between two inaccessible points by trigonometric levelling.
8. Theodolite Survey: Determining height of a remote object (inaccessible base).
9. Total Station: Setting up, orientation, measuring coordinates, area and perimeter computation using Total Station.
10. Total Station: Determination of distance between two inaccessible points, stake-out/setting out a structure.

11. Curve Setting: Setting out a simple curve using Rankine's deflection angle method.

12. Contour Mapping: Determining levels of contour points by grid method and plotting contour map.

13. AI Lab Exercise: Processing Total Station coordinate data using Python/QGIS to generate digital survey plan and compute areas.

14. Drone Survey Demonstration (or Virtual Lab): Observing drone flight plan, GCP placement, and processing photogrammetric output using Open Drone Map or WebODM.

AI Tools for Surveying Lab:

- QGIS (free, open-source) – Processing and visualizing survey data, generating contour maps from levelling data
- Python (geopandas, shapely, matplotlib) on Google Colab – Automated traverse plotting and area computation
- OpenDroneMap / WebODM (free) – Processing drone survey imagery into orthophotos and DEMs
- Cloud Compare (free) – Point cloud visualization from drone or LiDAR surveys
- Online Survey Calculators (free) – Quick verification of field computations

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01204P	PC	Strength of Materials Lab	0	0	0	3	1.5

Pre-Requisites: Basics of Strength of Materials and Engineering Mechanics

Course Objectives:

- To determine mechanical properties of engineering materials (E, G, yield strength, UTS, shear strength) using standard testing machines.
- To verify beam deflection theory for cantilever and simply supported beams in steel and wood under various loading conditions.
- To determine torsional properties of circular shafts and load–deflection characteristics of open-coiled and closed-coiled helical springs.
- To determine hardness numbers, impact strength, compressive strength of concrete specimens and verify continuous beam theory experimentally.
- To automate experimental data processing and develop AI-assisted material property analysis using Python and scikit-learn.

Course Outcomes (COs):

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

CO1: Conduct uniaxial tension, shear and impact tests on engineering material specimens using UTM and impact testing machine; determine mechanical properties including E, UTS, yield strength, shear strength, percentage elongation and impact energy; plot and interpret stress–strain and load–deflection diagrams. (L3)

CO2: Determine mid-span and multi-point deflections of steel and wood cantilever beams and simply supported beams under loading; verify theoretical bending relationships ($E = PL^3/3\delta I$ for cantilever; $E = PL^3/48\delta I$ for SS beam); compute E for each material and compare across specimen types. (L4)

CO3: Determine torsional properties of circular shafts (modulus of rigidity G, torque–twist relationship); measure stiffness and load–deflection characteristics of open-coiled and closed-coiled helical springs; verify spring stiffness formulae; experimentally determine deflections and verify continuous beam theory. (L4)

CO4: Determine Brinell / Rockwell hardness numbers; measure compressive strength and deformation characteristics of concrete specimens; conduct Izod and Charpy impact tests; compare ductile and brittle fracture behaviour; evaluate concrete deformation curve against theoretical stress–strain models. (L5)

CO5: Automate stress–strain curve plotting and property extraction using Python (NumPy / Matplotlib); develop ML-based material strength prediction using scikit-learn Random Forest regression; process deflection data and extract E by linear regression; evaluate model accuracy using R^2 and RMSE. (L6)

CO – PO Articulation Matrix:

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

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

List of Field Work/Lab Experiments:

Part A- Regular Experiments:

1. Determination of Mechanical Properties of Mild Steel under Uniaxial Tension
2. Determination of Deflection and Verification of Beam Theory for Steel Cantilever Beams
3. Determination of Deflection and Verification of Beam Theory for Wood Cantilever Beams
4. Determination of Deflection and Verification of Theoretical Bending Relationships for Wood Beams
5. Determination of Deflection and Verification of Theoretical Bending Relationships for Steel Beams
6. Determination of Torsional Properties of Circular Shafts
7. Determination of Hardness Number using Standard Hardness Testing Methods
8. Determination of Open-Coiled Helical Spring Stiffness and Load–Deflection Characteristics
9. Determination of Closed-Coiled Helical Spring Stiffness and Load–Deflection Characteristics
10. Determination of Compressive Strength and Deformation Characteristics Concrete Specimens
11. Determination of Impact Strength using Izod and Charpy Impact Tests
12. Determination of Shear Strength and Failure Characteristics
13. Experimental Determination of Deflections and Verification of Continuous Beam Theory

Part B- AI Integration Lab Exercises:

1. Stress–Strain Curve Automation (Python / Google Colab): input raw load–extension data from Experiment 1 (uniaxial tension test); compute engineering stress and strain; plot the stress–strain curve using Matplotlib; automate extraction of Young's modulus (E), 0.2% proof stress, UTS, percentage elongation and mark fracture point.
2. ML-Based Material Strength Prediction (scikit-learn): using a provided dataset of specimen dimensions, material grades and test parameters, train a Random Forest Regression model to predict UTS and hardness; evaluate performance using R^2 and RMSE; compare predicted values with experimental results from Experiments 1 and 7.
3. Deflection Data Analysis (Python / NumPy): input experimental load–deflection data from cantilever and simply supported beam experiments (Experiments 2–5 and 13); compute E by linear regression on δ vs. P; overlay the theoretical curve ($\delta = PL^3/3EI$ or $PL^3/48EI$); quantify percentage error for each specimen

AI Integration:

AI integration adds three exercises following the regular experiments: (i) Python automation of the stress–strain curve (Experiment 1) – plotting and automatic property extraction using Matplotlib/NumPy; (ii) scikit-learn Random Forest regression for predicting UTS and hardness from material and test parameters, validated against Experiment 1 and 7 results; (iii) Python deflection regression for Experiments 2–5 and 13 – fitting P vs. δ data, extracting E from slope, overlaying theoretical beam formula curve and computing percentage error – reinforcing the connection between the physical experiment and the analytical theory.

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

Recommended Free & Open-Source AI / Computational Tools Tools :

- Python (NumPy, Matplotlib, scikit-learn) on Google Colab (free) – stress–strain automation, deflection regression, ML strength prediction
- scikit-learn (free, open-source) – Random Forest regression for UTS and hardness prediction
- Virtual SOM Lab – VLabs IIT (free): https://soms-iitd.vlabs.ac.in – simulation of tension, torsion, hardness and impact tests
- LibreOffice Calc (free, open-source) – standardised data recording templates for all experiments
- GeoGebra (free) – interactive graphical verification of beam deflection profiles

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26SE01201S	SE	Building Planning and Drawing (AI Integrated)	1	0	0	2	2

Pre-Requisites: Fundamentals of Engineering Graphics and Basic civil Engineering

Course Objectives:

- To introduce students to national building bye-laws, zoning regulations, and planning principles for residential and public buildings.
- To impart drawing skills for sign conventions, bonds, doors, windows, and other building elements.
- To develop proficiency in preparing plan, elevation, and section drawings for various building types.
- To introduce Building Information Modelling (BIM) concepts and AI-assisted building design tools.
- To enable students to use free/open-source CAD and BIM software for building planning.

Course Outcomes (COs):

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

CO1: Apply national building bye-laws, sign conventions, and drawing standards to prepare accurate 2D drawings of building components including bonds, doors, windows, roofs, and staircases. (L3)

CO2: Apply the principles of residential and public building planning to prepare complete plan, elevation, and section drawings for single-storey residential, hospital, and industrial buildings in compliance with NBC 2016. (L3)

CO3: Analyze building design requirements and create 3D building models using BIM software (Revit/Free CAD), incorporating walls, floors, roofs, doors, windows, and generating automated sections and elevations. (L4)

CO4: Evaluate the application of AI and generative design tools for building planning, energy simulation, space optimization, and automatic compliance checking with building regulations. (L5)

CO5: Design a complete building project using BIM principles, integrating AI-assisted tools for generative floor plan creation, energy performance optimization, and BIM-based quantity take-off. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Building Bye-Laws, Regulations & Drawing Fundamentals:

Core Content: National Building Code 2016: purposes, importance. Building bye-laws: importance, types, setbacks, FAR, coverage area, height restrictions, parking norms. Sign conventions: walls, windows, doors, staircases, columns, materials. English Bond and Flemish Bond: advantages, applications, drawing. IS conventions for plan, elevation, and section drawings.

AI Integration: Introduction to digital building regulations: online NBC 2016 database access. AI in building regulation compliance: overview of automated zoning check systems. Using AutoCAD (student version) for basic 2D drawing conventions and sign conventions.

Tools: AutoCAD (free student version); LibreCAD (free, open-source 2D CAD); FreeCAD (free, open-source 3D CAD)

Unit II: Building Components & Detailing:

Core Content: Detailing and drawing of: doors (flush, panelled, glazed) – types, dimensions, IS standards; windows (casement, sliding, bay) – types, dimensions; ventilators, skylights; flat, sloping, and curved roofs – types, materials, elements; staircases – types, planning rules; lintels, arches, sunshades, chajjas.

AI Integration: BIM introduction: what is BIM and why it matters. Autodesk Revit (student version free) basics: inserting doors, windows, walls. AI-generated building component libraries. Parametric design concept: changing dimensions and auto-updating drawings. Introduction to generative design in architecture (concept only).

Tools: Autodesk Revit (free student version 2024); AutoCAD (free student); FreeCAD (free); SketchUp Free (online)

Unit III: Residential Building Planning:

Core Content: Principles of planning: aspect, prospect, privacy, furniture requirement, flexibility, roominess, sanitation, economy. Planning of residential buildings: types, room dimensions as per NBC, adjacency requirements. Line diagram preparation: applying building bye-laws to plot area. Drawing plan, elevation, and section of single storey residential building from line diagram.

AI Integration: AI-assisted floor plan generation: overview of tools like FloorPlanner AI (free), RoomGPT (free). Automated space planning: AI optimization for room adjacency and circulation. Using Autodesk Revit for 3D model creation of residential building from 2D plan. Virtual Reality (VR) walkthroughs using SketchUp and free VR viewer.

Tools: Autodesk Revit (free student); FloorPlanner.com (free online); SketchUp Free (online); AutoCAD student version

Unit IV: Public Buildings Planning & Drawing:

Core Content: Planning principles for public buildings: functional efficiency, accessibility (universal design, ramps, lifts), circulation, parking. Hospital building: departments, adjacency, infection control zones, planning and drawing. School/college building: classroom dimensions, corridors, toilets, library, planning and drawing. Industrial building: clear height, column grid, material movement, drawing.

AI Integration: BIM for public buildings: Level of Detail (LOD) concepts. AI for accessibility compliance checking. Universal Design principles and AI assistance. BIM 4D (scheduling) concept: linking construction activities to 3D model. Energy simulation of buildings using OpenStudio (free). Introduction to green building rating compliance using AI.

Tools: Autodesk Revit (free student); OpenStudio (free, NREL tool for energy simulation); AutoCAD; Design Builder (free student version for energy simulation)

Unit V: BIM Fundamentals & AI in Building Design:

Core Content: Introduction to BIM: definition, levels (0,1,2,3), standards (ISO 19650), benefits over 2D CAD. Revit interface: levels, grids, walls, floors, ceilings, roofs. Creating sections and elevations automatically from 3D model. Rendering and visualization. Introduction to clash detection: MEP vs. structure conflicts. BIM collaboration: IFC file format, interoperability.

AI Integration: AI in architecture and building design: generative design (Autodesk Revit add-ins). Automated quantity take-off from BIM model. Machine learning for energy optimization in building design. Smart buildings: IoT integration with BIM for facility management. Autodesk Generative Design (Fusion 360 free for education): concept. AI-based site suitability analysis.

Tools: Autodesk Revit 2024 (free student version); BIMcollab Zoom (free BIM viewer); FreeCAD (free BIM module); BlenderBIM (free, open-source BIM in Blender); IFC.js (open-source BIM viewer)

Text Books:

1. Gurcharan Singh and Jagdish Singh, Planning, Designing and Scheduling, Standard Publishers,

2019.

2. Chakraborti M., Building Planning and Drawing, Tata McGraw Hill, 2016.
3. Shah M.G., Kale C.M. and Patki S.Y., Building Drawing, Tata McGraw Hill, New Delhi.

Reference Books:

1. National Building Code of India 2016 (Vol. I & II), Bureau of Indian Standards.
2. Verma B.P., Civil Engineering Drawing and House Planning, Khanna Publishers, New Delhi.
3. Compendium of Architectural Norms & Guidelines for Educational Institutions – Central Public Works Department
4. Educational Building Norms Structural Engineering Drawings – Trimurty Premier Publishing House, Hyderabad

Online Resources:

1. Autodesk Revit 2024 – Free student version (autodesk.com/education)
2. AutoCAD 2024 – Free student version (autodesk.com/education)
3. FreeCAD with BIM Workbench – Fully free, open-source (freecadweb.org)
4. BlenderBIM – Free, open-source BIM tool in Blender (blenderbim.org)
5. LibreCAD – Free, open-source 2D CAD software (librecad.org)
6. SketchUp Free – Free online 3D modelling (sketchup.com/plans-and-pricing/sketchup-free)
7. OpenStudio – Free building energy simulation by NREL (openstudio.net)
8. FloorPlanner.com – Free AI-assisted floor plan creation (floorplanner.com)

B. Tech. – II Year I Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26BSHB214A	MC	**Environmental Science** (Common to all branches of Engineering)	2	0	0	0	0

Pre-Requisites: No specific prerequisites, open to all engineering branches.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply environmental concepts to assess resource exploitation and related environmental problems using Indian and global case studies. (L3)

CO2: Describe ecosystems, ecological succession, food chains, biodiversity, its threats, values, and conservation strategies, including India's mega-diversity status. (L3)

CO3: Identify pollution causes, effects and control measures; apply solid waste management principles and analyze pollution and disaster management case studies. (L4)

CO4: Analyze unsustainable development and climate-related issues; evaluate environmental impacts and Indian environmental legislation and its enforcement challenges. (L5)

CO5: Describe population–environment–health relationships and apply IT, AI and digital tools such as QGIS, Python and iNaturalist for environmental monitoring and documentation. (L3)

CO – PO Articulation Matrix: Scale: 3 – Strong, 2 – Moderate, 1 – Slight

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

Unit I: Multidisciplinary Nature of Environmental Studies and Natural Resources:

Core Content: Multidisciplinary Nature of Environmental Studies: – Definition, Scope and Importance – Need for Public Awareness.

Natural Resources: Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over-exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems – Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies – Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies – Energy resources: Use and exploitation of conventional and non-conventional energy sources, case studies.

AI Integration: AI in Natural Resource Monitoring: Remote sensing and satellite imagery analysis using Google Earth Engine (free cloud platform) for real-time deforestation detection and land use change mapping. Machine learning for groundwater level prediction from historical well data (Python/scikit-learn). AI-based precision agriculture – reducing fertilizer and pesticide use through crop health monitoring using drone imagery and computer vision (overview). ML forecasting of

renewable energy output (solar irradiance, wind speed) for energy resource planning.
Free / Open-Source AI Tools: Google Earth Engine (free cloud GIS); QGIS (free, open-source); Python (scikit-learn, Matplotlib) – Google Colab (free); Global Forest Watch (free dashboard).

Unit II: Ecosystems and Biodiversity and its Conservation:
Core Content: Ecosystems: Concept of an ecosystem – Structure and function of an ecosystem – Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of: Forest ecosystem; Grassland ecosystem; Desert ecosystem; Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries).
Biodiversity and its Conservation: Introduction – Definition: genetic, species and ecosystem diversity – Bio-geographical classification of India – Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values – Biodiversity at global, national and local levels – India as a mega-diversity nation – Hot-spots of biodiversity – Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts – Endangered and endemic species of India – Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.
AI Integration: AI in Ecosystem and Biodiversity Studies: AI-powered species identification – applications such as iNaturalist (free) use image recognition to identify plants, animals and insects from photographs, enabling citizen science biodiversity surveys. ML for biodiversity hotspot mapping using satellite imagery and species distribution models (MaxEnt – free). Computer vision for automated wildlife camera trap image analysis – identifying and counting species without manual review. Google Earth Engine for ecosystem health monitoring: NDVI time-series analysis for vegetation cover change detection. AI-assisted assessment of in-situ vs. ex-situ conservation effectiveness.
Free / Open-Source AI Tools: iNaturalist (free app/web); MaxEnt (free species distribution model); Google Earth Engine (free); QGIS (free); Python (rasterio, geopandas) – Google Colab.

Unit III: Environmental Pollution and Solid Waste Management:
Core Content: Environmental Pollution: Definition, cause, effects and control measures of: Air Pollution; Water Pollution; Soil Pollution; Marine Pollution; Noise Pollution; Thermal Pollution; Nuclear Hazards.
Solid Waste Management: Causes, effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster Management: floods, earthquake, cyclone and landslides.

AI Integration: AI in Pollution Monitoring and Disaster Management: ML-based Air Quality Index (AQI) prediction from meteorological data and traffic density (Python/scikit-learn/TensorFlow – free). IoT sensor networks with AI-driven anomaly detection for real-time water quality monitoring. Computer vision for automated solid waste sorting and classification (overview of AI-enabled sorting robots). OpenAQ platform (free, open data) for accessing global air quality sensor data for ML model training. AI and GIS (QGIS) for disaster risk zone mapping – flood inundation modelling, earthquake vulnerability assessment and cyclone track prediction.
Free / Open-Source AI Tools: OpenAQ (free global AQ data API); QGIS (free); Python (scikit-learn, TensorFlow, Matplotlib) – Google Colab (free); CPCB data portal (free).

Unit IV: Social Issues and the Environment:
Core Content: Social Issues and the Environment: From unsustainable to sustainable development – Urban problems related to energy – Water conservation, rain water harvesting, watershed management – Resettlement and rehabilitation of people; its problems and concerns, case studies – Environmental ethics: issues and possible solutions – Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust, case studies – Wasteland reclamation – Consumerism and waste products – Environment Protection Act – Air (Prevention and Control of Pollution) Act – Water (Prevention and Control of Pollution) Act – Wildlife Protection Act – Forest Conservation Act – Issues involved in enforcement of environmental legislation – Public awareness.
AI Integration: AI for Climate Change and Sustainable Development: Global climate model data analysis using Python (xarray, Matplotlib) from freely available CMIP6 datasets – visualising temperature rise, precipitation change and sea level scenarios. ML for carbon footprint estimation

from energy consumption and transport data (Python/scikit-learn). AI-based watershed management: QGIS + Python for delineation, runoff modelling and rainwater harvesting potential mapping. Overview of AI in environmental compliance monitoring – automated identification of violations using satellite data. Digital tools for public environmental awareness: social media sentiment analysis for pollution perception studies (Python/NLP).

Free / Open-Source AI Tools: Python (xarray, Matplotlib, scikit-learn) – Google Colab (free); CMIP6 climate data (free, ESGF portal); QGIS (free); openLCA (free, LCA software).

Unit V: Human Population and the Environment:

Core Content: Human Population and the Environment: Population growth, variation among nations – Population explosion – Family Welfare Programmes – Environment and human health – Human Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of Information Technology in Environment and human health – Case studies.

Field Work: Visit to a local area to document environmental assets: river / forest / grassland / hill / mountain – Visit to a local polluted site: Urban / Rural / Industrial / Agricultural – Study of common plants, insects and birds – river, hill slopes, etc.

AI Integration: AI in Population Studies and Public Health: ML-based population growth forecasting from census data using Python (time-series regression with scikit-learn / LSTM with TensorFlow). AI in public health surveillance – disease outbreak prediction and mapping using spatiotemporal ML models (overview of WHO FluNet, IDSP data). Role of Information Technology and AI in environmental health: wearable sensors for personal air quality monitoring;

Smartphone-based water quality testing with ML image analysis. Citizen science apps for field work data collection: iNaturalist, eBird (free) for recording plants, birds and insects during field visits; OpenStreetMap for documenting environmental assets. GIS-based environmental health mapping using QGIS (free).

Free / Open-Source AI Tools: iNaturalist (free); eBird (free); QGIS (free); Python (scikit-learn, TensorFlow) – Google Colab (free); OpenStreetMap (free).

Text Books:

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

Reference Books:

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

Online Resources:

1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.coursera.org/learn/python?specialization=python#syllabus
3. NPTEL: Environmental Science – https://nptel.ac.in/courses/124107002
4. Google Earth Engine (free cloud geospatial platform) – https://earthengine.google.com
5. Global Forest Watch (free deforestation monitoring) – https://www.globalforestwatch.org
6. OpenAQ (free global air quality open data) – https://openaq.org
7. iNaturalist (free citizen science biodiversity app) – https://www.inaturalist.org

8. QGIS (free, open-source GIS) – https://qgis.org
9. India Meteorological Department (free data) – https://mausam.imd.gov.in
10. Central Pollution Control Board (CPCB) data portal (free) – https://cpcb.nic.in

Recommended Free & Open-Source AI / Digital Tools:

1. Google Earth Engine (free cloud platform) – real-time satellite data analysis for deforestation, land use and vegetation change detection
2. QGIS (free, open-source GIS) – environmental mapping, watershed delineation, disaster risk zone mapping, biodiversity hotspot analysis
3. Python (NumPy, Pandas, Matplotlib, scikit-learn, TensorFlow) on Google Colab (free) – AQI prediction, population growth ML models, climate data analysis
4. iNaturalist (free, open-source citizen science) – species identification using image recognition during field work; community biodiversity database
5. eBird (free, Cornell Lab) – bird species recording and mapping during field visits; global bird distribution data
6. OpenAQ (free, open-source) – global real-time and historical air quality data for ML model training
7. MaxEnt (free) – species distribution modelling for biodiversity hotspot prediction from occurrence data
8. Global Forest Watch (free dashboard) – monitoring deforestation and forest cover change using satellite imagery
9. CMIP6 climate datasets (free, ESGF portal) – global climate model data for Python-based climate change scenario analysis

B. Tech. – II Year
II Semester

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26BSHB213T	BS	**Biology for Engineers** (Common to all branches of Engineering)	2	0	0	0	2

Pre-Requisites: Basic Sciences (Intermediate level)

Course Objectives:

- To provide foundational biological knowledge with engineering relevance.
- To demonstrate AI-driven applications in biology and bioengineering.
- To encourage interdisciplinary teaching that connects biology, engineering, and computational tools.
- To prepare students for innovation in healthcare, materials, and bio-industries.
- To evaluate bio mimetic principles and emerging bioengineering trends for real-world engineering applications.

Course Outcomes (COs):

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

CO1: Apply the fundamentals of biology, biomolecules, and human systems to interpret their relevance in engineering applications. (L3)

CO2: Apply AI tools and computational techniques for biological data analysis, bio molecule visualisation, and healthcare applications. (L3)

CO3: Analyse bio molecules and bio-inspired systems for solving engineering problems and developing sustainable technologies. (L4)

CO4: Design bio mimetic and bioengineering solutions using interdisciplinary approaches and AI-enabled tools. (L5)

CO5: Evaluate emerging trends in bioengineering such as 3D bio printing, regenerative medicine, and AI-based diagnostics for societal applications. (L5)

CO – PO Articulation Matrix:

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

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

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 → super hydrophobic 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

Text Books:

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

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.

Online Resources:

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

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26HMHS204T	HM	**Universal Human Values – Understanding Harmony and Ethical Human Conduct** (Common to all branches of Engineering)	3	0	0	0	3

Pre-Requisites: Basic understanding of ethics, human values, society and environmental responsibility, along with openness to self-reflection and value-based learning.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the concepts of universal human values, natural acceptance, and the purpose of human life to examine personal relationships, family harmony, and societal situations. (L3)

CO2: Apply the framework of values-based living to personal relationships, family harmony, and societal well-being in daily life contexts. (L3)

CO3: Analyse ethical dilemmas in engineering practice using the lens of universal human values and differentiate value-driven from interest-driven conduct. (L4)

CO4: Evaluate current engineering and social practices for their alignment with sustainability, ecological balance, and universal human order. (L5)

CO5: Create a personal value charter and an action plan for ethical engineering practice using AI reflection tools and collaborative dialogue. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Introduction to Value Education and Human Consciousness:

Core Concepts: Need for Value Education: Crisis of values in modern society, Human Being: A co-existence of Self (Jeevan) and Body; Consciousness as a distinguishing feature, Natural Acceptance: The innate human capacity to know what is right, Activities of the Self: Knowing, Assuming, Recognising, Fulfiling, Difference between Animal Consciousness and Human Consciousness

AI Integration: AI Tool: ChatGPT – present ethical dilemmas to AI and compare AI responses with naturally accepted human values, **Reflection Activity:** Use an AI journaling assistant (Day One free, or Notion AI) to document daily natural acceptances and violations, Discussion: Can AI ever have human values? Debate using structured arguments

Unit II: Harmony in the Individual:

Core Concepts: Harmony between Self and Body: Needs of the Self (happiness, trust, etc.) 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: AI Tool: Replika (free AI companion) – explore conversations about emotional needs; critically reflect on AI limitations in understanding values, Activity: Create a 'Well-being Wheel' using Canva; use AI to identify gaps between material and value-based happiness, Journaling: Write a weekly reflection on one feeling (e.g., trust) experienced and observed

Unit III: Harmony in the Family and Society:
Core Concepts: 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: AI Tool: ChatGPT – generate scenarios of family conflicts and analyse through the lens of trust and respect, Case Study with AI: Use AI to map historical social movements (Gandhi, Ambedkar) onto the five-order framework, Project: Document 3 community service observations and analyse them using the harmony framework; use AI for structuring the report

Unit IV: Harmony in Nature and Existence:
Core Concepts: Four Orders of Nature: Material, Plant/Bio, Animal, Human, Interconnectedness: Mutual fulfilment among the four orders, Ecological Balance: Current ecological crises and human responsibility, Sustainable Development Goals (SDGs) and Value-based Engineering, Right Understanding of Existence: Co-existence of units in space
AI Integration: AI Tool: Global Forest Watch (open-source platform) – analyse deforestation data and correlate with lack of ecological values, AI Tool: ChatGPT – generate a sustainable engineering project idea and critique it using ecological harmony principles, Activity: Create an ecological audit of the campus using satellite images (Google Earth) and document AI-assisted analysis

Unit V: Ethical Human Conduct and Professional Ethics:
Core Concepts: 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: AI Tool: IBM AI Fairness 360 (open-source) – demonstrate AI bias using sample datasets, Case Study: Analyse Bhopal Gas Tragedy / Challenger Disaster through professional ethics lens, AI Ethics Audit: Use a given AI system (chatbot, recommendation engine) and evaluate it on fairness and transparency, Capstone: Write a 'Personal Value Charter' as a future engineer; use ChatGPT for drafting and self-editing for authenticity

Text Books:

1. T1. Gaur, R.R., Sangal, R. &Bagaria, G.P. (2019). A Foundation Course in Human Values and Professional Ethics (3rd ed.). Excel Books, New Delhi.
2. T2. Nagraj, A. (2015). Human Values and Professional Ethics: Universal Human Order. Jeevan Vidya Prakashan, Amarkantak.

Reference Books:

1. R1. Subramanian, C.V. (2020). Professional Ethics and Human Values. Oxford University Press.

2. R2. Bajpai, Sarvesh Chandra. (2019). Engineering Ethics: Concepts and Cases. McGraw-Hill Education.
3. R3. Flood, R.L. (2018). Rethinking the Fifth Discipline: Learning within the Unknowable. Routledge.
4. R4. Covey, Stephen R. (2020). The 7 Habits of Highly Effective People (30th Anniversary ed.). Simon & Schuster.
5. R5. Krishnamurti, J. (2018). Education and the Significance of Life. Harper Collins.
6. R6. Singer, Peter. (2021). Practical Ethics (4th ed.). Cambridge University Press.

Online Resources:

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

YouTube / Video Resources:

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

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01205T	PC	Fluid Mechanics (AI Integrated)	2	0	2	0	3

Pre-Requisites: Basics of Mathematics (Calculus, Vector Calculus), Engineering Mechanics and Basic Physics

Course Objectives:

- To understand the fundamental properties of fluids and their classification including viscosity, surface tension, and compressibility.
- To apply fluid statics principles to compute hydrostatic pressure, forces on surfaces, and buoyancy.
- To analyze different types of fluid flows using kinematics concepts including stream functions and velocity potential.
- To bernoulli's equation and momentum principle to flow measurement devices and hydraulic systems.
- To integrate AI/ML and CFD tools for fluid flow analysis, optimization of hydraulic designs, and prediction of flow behaviour.

Course Outcomes (COs):

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

CO1: Apply the properties of fluids, including viscosity, surface tension, and bulk modulus, and concepts of fluid flow to solve basic civil and environmental engineering problems(L3)

CO2: Apply the laws of fluid statics (Pascal's law, hydrostatic pressure, buoyancy, stability) to calculate hydrostatic forces on plane and curved surfaces and evaluate the stability of floating structures. (L3)

CO3: Analyze fluid kinematic phenomena using stream functions, velocity potential, and continuity equation, and differentiate between types of flows, interpreting CFD-based visualizations of flow fields. (L4)

CO4: Evaluate the application of Bernoulli's equation, momentum principle, and dimensionless numbers to flow measurement devices and hydraulic systems, and compare experimental/CFD results with analytical solutions. (L5)

CO5: Design computational pipe network solutions using EPANET and Python, develop ML models for flow prediction and leakage detection, and propose AI-enhanced hydraulic monitoring systems for civil water infrastructure. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Properties of Fluids:

Core Content: Distinction between fluid and solid; types of fluids: ideal, real, Newtonian, non-Newtonian. Fluid properties: density, specific weight, specific gravity. Viscosity: dynamic and kinematic viscosity; Newton's law of viscosity; variation with temperature; measurement. Vapour pressure, boiling point, cavitation. Surface tension, capillarity. Bulk modulus of elasticity, compressibility. Pressure: absolute, gauge, vacuum, and atmospheric pressure.

AI Integration: ML for fluid property prediction: using regression models to predict viscosity from temperature and composition data. Python-based data analysis of fluid properties using Pandas. Introduction to AI in fluid mechanics: how ML is used in computational fluid dynamics (CFD) to accelerate simulations. Overview of Physics-Informed Neural Networks (PINNs) for fluid mechanics (conceptual).

Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib, Pandas) – Google Colab; NIST WebBook (free database for fluid properties); Scilab (free)

Unit II: Fluid Statics:

Core Content: Pressure at a point: Pascal's law; pressure variation with depth; concept of pressure head. Pressure measurement: piezometers, U-tube manometers, differential manometers, mechanical gauges. Hydrostatic forces on plane surfaces: horizontal, vertical, and inclined; centre of pressure. Hydrostatic forces on curved surfaces: component method. Buoyancy and stability of floating bodies: metacentric height, conditions of stability. Introduction to fluid statics in engineering applications.

AI Integration: Python/MATLAB simulation of hydrostatic pressure distribution. AI for dam safety: ML-based prediction of uplift pressure on dam foundations. CFD software introduction: OpenFOAM (free) for visualizing pressure distribution in tanks. Automated manometer reading using image processing (computer vision with OpenCV – concept).

Free/Open-Source AI Tools: OpenFOAM (free, open-source CFD); Python (NumPy, Matplotlib) – Google Colab; online hydrostatic calculators (free); ANSYS Fluent student version (free)

Unit III: Fluid Kinematics:

Core Content: Classification of fluid flow: steady/unsteady, uniform/non-uniform, laminar/turbulent, rotational/irrotational, compressible/incompressible. Reynolds number: significance. Streamlines, pathlines, streaklines, stream tubes. Stream function: properties, Stokes stream function. Velocity potential function: irrotational flow, potential flow. Continuity equation: derivation in Cartesian coordinates (1D, 2D, 3D). Introduction to Navier-Stokes equations (concept).

AI Integration: AI-based flow visualization: using ML for particle image velocimetry (PIV) data processing. Python simulation of potential flow around a cylinder. Introduction to CFD: setting up a simple laminar flow problem in OpenFOAM. AI in turbulence modelling: overview of data-driven turbulence models. Visualizing streamlines using Python (matplotlib streamplot).

Free/Open-Source AI Tools: OpenFOAM (free, open-source CFD); Python (NumPy, Matplotlib) – Google Colab; SALOME (free pre/post-processing for OpenFOAM); SimScale (free cloud CFD with limited hours)

Unit IV: Fluid Dynamics – Bernoulli's Equation & Flow Measurement:

Core Content: Surface and body forces; Euler's equation of motion; Bernoulli's equation: derivation, assumptions, practical applications. Energy equation. Venturimeter: principle, coefficient of discharge, calibration. Orifice meter, Pitot tube. Momentum principle: impulse-momentum equation; forces on pipe bends; water hammer (introduction). Vortex flow: free and forced vortex. Dimensionless numbers: Reynolds, Froude, Mach, Weber, Euler numbers.

AI Integration: ML for flow meter calibration: predicting coefficient of discharge from Reynolds number using regression. Python simulation of Bernoulli's theorem with losses. CFD simulation of Venturimeter flow using OpenFOAM (simplified tutorial). AI for real-time flow monitoring: IoT sensor integration with ML for flow prediction. Virtual lab for Bernoulli's theorem experiment.

Free/Open-Source AI Tools: OpenFOAM (free CFD); Python (SciPy, NumPy) – Google Colab; EPANET (free, USEPA water distribution network software); virtual lab simulations (VLAB, free)

Unit V: Pipe Flow Analysis & AI Applications:
Core Content: Laminar flow in pipes: Hagen-Poiseuille equation; velocity distribution. Turbulent flow: velocity distribution; boundary layer theory (introduction). Darcy-Weisbach equation for head loss. Moody's diagram: friction factor determination. Minor losses: sudden expansion, contraction, bends, valves. Hydraulic Grade Line (HGL) and Total Energy Line (TEL). Pipes in series and parallel; equivalent length; branching pipes (three-reservoir problem). Introduction to water hammer.
AI Integration: ML for pipe network optimization: using Python to solve pipe flow problems and optimize pipe diameters. AI for leakage detection in water distribution networks: anomaly detection using sensor data. CFD for pipe fitting loss coefficient prediction using OpenFOAM. EPANET Python toolkit (EPyT, free) for automated water network analysis. ML-based water demand forecasting for municipal water supply.
Free/Open-Source AI Tools: EPANET (free, USEPA) and EPyT Python wrapper (free, open-source); OpenFOAM (free CFD); Python (scikit-learn, NumPy) – Google Colab; WaterGEMS student version

Text Books:

1. Modi P.M. and Seth S.M., Hydraulics and Fluid Mechanics, Standard Book House, 22nd Edition, 2019.
2. Subrahmanya K., Theory and Applications of Fluid Mechanics, Tata McGraw Hill, 2nd Edition, 2018.

Reference Books:

1. Bansal R.K., A Textbook of Fluid Mechanics and Hydraulic Machines, Laxmi Publications, 11th Edition, 2024.
2. White F.M. and Xue H., Fluid Mechanics, Tata McGraw Hill, 9th Edition, 2022.
3. Cengel Y.A. and Cimbala J.M., Fluid Mechanics: Fundamentals and Applications, McGraw Hill, 4th Edition, 2017.
4. Munson B.R., Young D.F. and Okiishi T.H., Fundamentals of Fluid Mechanics, Wiley, 7th Edition, 2012.
5. Pillai N.N., Principles of Fluid Mechanics and Fluid Machines, Universities Press, 3rd Edition, 2009.

Online Resources:

1. NPTEL: Fluid Mechanics – https://archive.nptel.ac.in/courses/112/105/112105269
2. NPTEL: Fluid Mechanics and Hydraulics – https://nptel.ac.in/courses/112104118
3. OpenFOAM Official Documentation (free CFD): https://openfoam.org
4. EPANET (free water network analysis): https://www.epa.gov/water-research/epanet
5. SimScale (free cloud CFD): https://www.simscale.com
6. MIT OCW Fluid Mechanics: https://ocw.mit.edu/courses/2-20-marine-hydrodynamics
7. Google Colab for Python CFD tutorials: https://colab.research.google.com

Recommended Free & Open-Source AI/Computational Tools:

1. OpenFOAM (free, open-source) – Industry-standard CFD for pipe flow, Venturimeter, and fluid dynamics
2. EPANET 2.2 (free, US EPA) – Water pipe network analysis; EPyT (free Python wrapper)
3. Python (NumPy, SciPy, Matplotlib) on Google Colab – Numerical fluid mechanics solutions
4. SimScale (free cloud tier) – Web-based CFD simulation for simple flow problems
5. Scilab (free) – MATLAB-alternative for fluid property and flow calculations
6. SALOME (free, open-source) – Pre/post processing for OpenFOAM CFD
7. ParaView (free, open-source) – CFD result visualization

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01206T	PC	Concrete Technology (AI Integrated)	2	0	2	0	3

Pre-Requisites: Basics of Engineering Chemistry, Engineering Mechanics, Basic Materials Knowledge

Course Objectives:

- To understand the properties, functions, and testing of cement, aggregates, and admixtures as ingredients of concrete.
- To evaluate the properties of fresh concrete and their influence on workability, placement, and quality.
- To analyze the mechanical and durability properties of hardened concrete including compressive strength, flexure, tension, and NDT methods.
- To understand the effects of creep, shrinkage, and elasticity on concrete structural behaviour.
- To apply AI/ML tools for mix design optimization, quality control, strength prediction, and durability assessment of concrete.

Course Outcomes (COs):

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

CO1: Apply knowledge of cement composition, hydration, aggregate properties, and mineral and chemical admixtures to select and assess concrete ingredients for concrete production. (L3)

CO2: Apply standard test procedures to evaluate workability, segregation, and bleeding of fresh concrete, and assess the influence of mix proportions, temperature, and admixtures on fresh concrete properties. (L3)

CO3: Analyze the mechanical properties of hardened concrete including compressive, flexural, and split tensile strength, and evaluate NDT test results (UPV, rebound hammer) to assess in-situ concrete quality using ML-based prediction models. (L4)

CO4: Evaluate the time-dependent deformation behaviour of concrete (creep and shrinkage) using analytical models and predictive ML algorithms, and assess their structural implications for serviceability and durability. (L5)

CO5: Design concrete mixes using IS 10262:2019 and ACI methods, implement AI/ML-based mix design optimization in Python, and develop automated quality control frameworks for special concrete types including HPC, SCC, and FRC. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Cement & Aggregates:

Core Content: Portland cement: chemical composition (Bogue compounds), hydration reactions, heat of hydration. Setting of cement: initial and final. Fineness, consistency, soundness, strength tests. Different grades of OPC (33, 43, 53) and types of cement (PPC, PSC, SRPC, etc.) and their applications. Admixtures: mineral admixtures – fly ash (IS 3812), GGBS, silica fume, metakaolin; chemical admixtures – accelerators, retarders, water reducers (plasticizers, superplasticizers), air entraining agents. Aggregates: classification, particle shape and texture; mechanical properties – impact, abrasion, crushing value; specific gravity, bulk density, porosity, absorption, moisture content; bulking of sand; deleterious substances; soundness; alkali-aggregate reaction; thermal properties. Sieve analysis, fineness modulus, grading curves. Quality of mixing water (IS 456).

AI Integration: ML for cement strength prediction: linear regression and neural network models predicting 28-day strength from mix data (Python/scikit-learn). AI for cement quality control: statistical process control using Python. Computer vision for aggregate shape classification using image processing (OpenCV). Introduction to materials informatics: data-driven discovery of new supplementary cementitious materials.

Free/Open-Source AI Tools: Python (scikit-learn, NumPy, Pandas, OpenCV) – Google Colab; ConcreteMix online calculator (free); LibreOffice Calc for sieve analysis

Unit II: Fresh Concrete:

Core Content: Steps in concrete manufacture: batching, mixing, transporting, placing, compacting, curing. Workability: definition, factors affecting workability, measurement methods – slump test, compaction factor test, Vee-Bee test, flow table test (IS codes). Segregation: causes, prevention. Bleeding: causes, effects, prevention. Setting time of concrete. Effect of time and temperature on workability. Ready-mixed concrete (RMC): types (transit-mixed, dry-batched), standards (IS 4926). Shotcrete (dry and wet process).

AI Integration: Computer vision for slump measurement: automated slump height detection using image processing. ML for workability prediction from mix proportions. AI-driven concrete batching plant automation: optimizing water-cement ratio in real time. IoT sensors for fresh concrete monitoring (temperature, humidity). Introduction to digital concrete quality management systems.

Free/Open-Source AI Tools: Python (OpenCV, scikit-learn) – Google Colab; virtual concrete lab simulations (free online); Python Pandas for mix design data analysis

Unit III: Hardened Concrete:

Core Content: Water-cement ratio: Abrams' law, gel/space ratio theory. Nature of strength: factors affecting compressive strength. Maturity concept. Relationship between compressive and tensile/flexural strength. Curing: methods, importance, duration. Testing of hardened concrete: Cube Strength, Cylindrical Strength, Flexural Strength ;compression test, flexure test (IS 516), split tensile test (IS 5816). Non-destructive testing (NDT): rebound hammer (IS 13311 Part 2), ultrasonic pulse velocity test (IS 13311 Part 1), core testing (IS 516), pull-out test. IS code provisions for NDT interpretation.

AI Integration: ML for concrete compressive strength prediction: ensemble methods (Random Forest, XGBoost) using mix design parameters as input features (Python/scikit-learn). Computer vision for crack detection in concrete: CNN-based image classification using pre-trained models (TensorFlow, free on Google Colab). AI for NDT data interpretation: ML to predict in-situ strength from UPV and rebound hammer readings. Automated concrete quality classification system.

Free/Open-Source AI Tools: Python (scikit-learn, TensorFlow, XGBoost) – Google Colab; Crack detection pre-trained models on TensorFlow Hub; virtual UPV and rebound hammer simulations (VLAB)

Unit IV: Elasticity, Creep & Shrinkage:

Core Content: Modulus of elasticity: static and dynamic (ultrasonic pulse velocity method). Poisson's ratio of concrete. Creep: definition, causes, factors influencing creep, creep coefficient, relationship between creep and time, specific creep, effects of creep on structures (prestress loss, deflection, redistribution). Shrinkage: types – plastic shrinkage, drying shrinkage, autogenous shrinkage, carbonation shrinkage. Factors affecting shrinkage. Differential shrinkage. Effects of shrinkage on structures: cracking, loss of prestress.

AI Integration: ML for creep prediction: ML models using ACI and fib model inputs (Python). Predictive modelling of concrete shrinkage using regression and neural networks. Python simulation of time-dependent concrete deformation (creep + shrinkage curves). Sensitivity analysis of creep factors using ML feature importance (scikit-learn). Introduction to digital twins for long-term monitoring of concrete bridge creep.
Free/Open-Source AI Tools: Python (scikit-learn, NumPy, Matplotlib) – Google Colab; DIANA FEA student version; ConcreteWorks (free, NRMCA – concrete thermal analysis)

Unit V: Mix Design, Special Concretes & AI Optimization:
Core Content: Mix design: objectives, factors, IS method (IS 10262:2019), ACI method. Statistical methods in quality control: standard deviation, coefficient of variation, acceptance criteria. Types of special concretes: high-performance concrete (HPC), self-compacting concrete (SCC), fibre-reinforced concrete (FRC) – steel, polypropylene, glass fibres; lightweight concrete, high-density concrete, roller-compacted concrete. Self-healing concrete: concept, mechanisms. Fibre reinforced concrete: properties, applications.
AI Integration: AI-optimized concrete mix design: multi-objective optimization using Python (minimize cement content, achieve target strength and workability). ML-based mix design from database of existing concrete mixes (neural network approach). Automated slump/strength acceptance using real-time sensor data and ML classifiers. Introduction to generative AI for novel concrete formulations. Python implementation of IS 10262:2019 mix design automated calculation.
Free/Open-Source AI Tools: Python (SciPy, scikit-learn, TensorFlow) – Google Colab – Complete mix design automation; ConcreteMix Optimizer (Python script, open source GitHub); Mix design spreadsheets (LibreOffice Calc, free); OpenMDAO (NASA, free) – Multi-objective mix design optimization

Text Books:

1. Gambhir M.L., Concrete Technology, Tata McGraw Hill, 5th Edition, 2013.
2. Thomas J., Concrete Technology, Cengage Publications, 1st Edition, 2015.
3. Concrete Technology, Shetty M.S, S Chand & Co, 2004

Reference Books:

1. Mehta P.K. and Monteiro P.J.M., Concrete: Microstructure, Properties and Materials, McGraw Hill, 4th Edition, 2014.
2. Brooks J.J. and Neville A.M., Concrete Technology, Pearson, 2nd Edition, 2019.
3. Neville A.M., Properties of Concrete, Pearson, 4th Edition (Indian Reprint).
4. IS 456:2000 – Plain and Reinforced Concrete, BIS, New Delhi.
5. IS 10262:2019 – Guidelines for Concrete Mix Design, BIS, New Delhi.
6. IS 13311 Part 1 & 2 – Non-Destructive Testing of Concrete, BIS.

Online Resources:

1. NPTEL: Concrete Technology – https://nptel.ac.in/courses/105104098
2. Virtual Labs Concrete Technology – https://ced-nitk.vlabs.ac.in
3. IS Codes Database – https://bis.gov.in
4. ConcreteWorks (free thermal analysis tool): https://concreteworks.com
5. Google Colab for ML-based mix design: https://colab.research.google.com
6. TensorFlow Hub for pre-trained models: https://tfhub.dev
7. NRMCA Mix Design Calculator: https://www.nrmca.org

Recommended Free & Open-Source AI/Computational Tools:

1. Python (scikit-learn, TensorFlow, XGBoost, NumPy) on Google Colab – ML for strength prediction and mix design optimization
2. OpenCV (free, open-source) on Google Colab – Image processing for crack detection and aggregate analysis
3. TensorFlow/Keras (free) on Google Colab – CNN for concrete crack detection using pre-trained models
4. LibreOffice Calc (free) – Mix design spreadsheets and statistical quality control
5. ConcreteWorks (free, NRMCA) – Concrete thermal analysis for mass concrete
6. Python SciPy (free) – Optimization for multi-objective mix design

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01207T	PC	Structural Analysis (AI Integrated)	2	0	2	0	3

Pre-Requisites: Basics of Strength of Materials, Engineering Mechanics, Engineering Mathematics

Course Objectives:

- To understand the principles of strain energy and apply energy theorems to analyze determinate and indeterminate structures.
- To apply force method (flexibility method) and displacement method (stiffness method) to analyze indeterminate structures.
- To understand the fundamental concepts of arches, including three-hinged and two- hinged arches, and analyze the effects of horizontal thrust, bending moment, normal thrust, and radial shear
- To apply slope-deflection and moment distribution methods to analyze continuous beams and portal frames.
- To integrate AI/ML tools and structural analysis software for automated structural analysis, design verification, and structural optimization.

Course Outcomes (COs):

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

CO1: Apply strain energy principles, Castigliano, Maxwell and Betti energy theorems, and the virtual work method to determine structural response and assess the indeterminacy of structural systems. (L3)

CO2: Apply the compatibility method (force/flexibility) and unit load method to analyze statically indeterminate beams and trusses, and compute deflections, reactions, and internal forces in indeterminate structural systems. (L3)

CO3: Explain the behaviour of three-hinged and two-hinged arches and analyze the effects of horizontal thrust, bending moment, normal thrust, and radial shear. (L4)

CO4: Evaluate the structural response of continuous beams and portal frames using slope-deflection and moment distribution methods considering support settlements and sway, and compare results with computational models (Python/OpenSees). (L5)

CO5: Design automated structural analysis workflows using Python programming (NumPy, SymPy) and FEM software (OpenSees, SAP2000), and propose AI/ML-based frameworks for structural response prediction, verification, and design optimization. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Energy Theorems & Virtual Work:

Core Content: Strain energy: in axial, bending, shear, and torsion. Castigliano's first and second theorems. Virtual work principle: unit load method (method of virtual work). Application to deflections of beams and pin-jointed trusses. Betti's law and Maxwell's reciprocal theorem. Introduction to computer methods in structural analysis: stiffness and flexibility approaches.

AI Integration: Python implementation of virtual work method for truss deflection computation. Introduction to matrix structural analysis: element stiffness matrix concept. Using Python/NumPy for matrix operations in structural analysis. OpenSees Python introduction: setting up a simple truss problem. Overview of commercial structural analysis software and how they implement energy methods internally.

Free/Open-Source AI Tools: Python (NumPy, SciPy) – Google Colab; OpenSees (free, Python interface); SkyCiv (free structural analysis online)

Unit II: Analysis of Indeterminate Structures:

Core Content: Degree of static indeterminacy: internal and external; degree of kinematic indeterminacy (DOF). Methods of analysis: compatibility (force/flexibility) method and equilibrium (stiffness/displacement) method. Analysis of propped cantilevers and two-span continuous beams using compatibility method. Analysis of statically indeterminate trusses: up to two degrees of indeterminacy using unit load method and Castigliano's second theorem. Stiffness method: concept, element stiffness matrix for bar element.

AI Integration: Python-based automated indeterminacy calculation and analysis for 2D frames. Introduction to matrix stiffness method: Python implementation of global stiffness assembly for beams. OpenSees Python: analysis of a propped cantilever with settlement. AI-assisted structural model verification: using ML to predict reactions and check equilibrium.

Free/Open-Source AI Tools: Python (NumPy, SciPy) – Google Colab; OpenSees (free, open-source); SAP2000 student version (free); SkyCiv free structural calculator

Unit III: Analysis of Three-Hinged and Two-Hinged Arches:

Core Content :Three-Hinged Arches: Introduction to arches: types (three-hinged, two-hinged, fixed); Eddy's theorem – statement and geometrical interpretation; determination of horizontal thrust (H), bending moment (BM), normal thrust (N) and radial shear (S) at any section of a three-hinged parabolic and circular arch under concentrated loads, UDL, and UVL; three-hinged arch with supports at different levels.

Two-Hinged Arches – Elastic Theory: Degree of static indeterminacy; derivation of horizontal thrust H using the principle of least work / Castigliano's second theorem; parabolic arch under full UDL – funicular condition (zero BM); determination of H, BM, N and S for two-hinged parabolic and circular arches under standard loading. Fixed Arches (No analytical questions in the Examination).

Effect of Temperature: Horizontal thrust due to rise or fall of temperature in a two-hinged arch – derivation of $H_t = EI\alpha T / \int(y^2/I)\,ds$; combined effect of loading and temperature change on H, BM, N and S.

Rib Shortening and Temperature Stresses: Additional horizontal thrust due to elastic shortening of arch rib under axial compression; combined expression for total H accounting for rib shortening and temperature; numerical problems. Integration

AI Integration: :**Python automation of arch analysis:** computing H, BM, N and S at multiple sections for three-hinged and two-hinged arches using NumPy; plotting BM diagrams, thrust lines and influence lines for H via Matplotlib. SciPy numerical integration (scipy.integrate.quad) for evaluating $\int(My\,ds/EI)$ and $\int(y^2\,ds/EI)$ in the elastic theory – enabling analysis of non-standard arch profiles beyond textbook cases.ML for **Horizontal Thrust Prediction:** Random Forest regression model (scikit-learn) trained on arch configuration parameters (span, rise, loading, temperature change) to predict H instantly for preliminary design; SciPy optimisation for funicular form-finding– the computational analogue of Eddy's theorem.

FEM validation: OpenSees Python interface for two-hinged and three-hinged arch modelling using curved beam elements with hinge releases; PyNite (open-source) for extracting M, N, S diagrams and validating elastic theory results; SkyCiv free solver for quick verification.

Free / Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib, scikit-learn) – Google Colab (free); OpenSees (free, open-source) with Python API; PyNite (free, open-source Python FEM); SkyCiv free structural analysis (online); SAP2000 student version (free).

Unit IV: Slope-Deflection Method:
Core Content: Assumptions and derivation of slope-deflection equations. Application to continuous beams: without settlement (symmetric and asymmetric loading), with settlement of one or more supports. Analysis of single-bay single-storey portal frames: without sway (symmetric frame, symmetric loading), with sway – general procedure, sway corrections. Sidesway analysis. Near and far end conditions.
AI Integration: Python implementation of slope-deflection method for continuous beams (automated solution of simultaneous equations using NumPy.linalg). Matrix formulation of slope-deflection equations. OpenSees verification of slope-deflection results. Introduction to AI for automated portal frame design: overview. Comparison of analytical and numerical (FEM) results using SAP2000 student version.
Free/Open-Source AI Tools: Python (NumPy, SymPy, Matplotlib) – Google Colab; OpenSees (free); SAP2000 student version (free); SkyCiv free frame calculator

Unit V: Moment Distribution Method & AI in Structural Analysis:
Core Content: Introduction to moment distribution method (Hardy Cross method). Distribution factors: stiffness factor, carry-over factor, derivation for near and far conditions. Unlocking and locking cycle. Application to continuous beams: without and with settlement of supports. Application to single-bay single-storey portal frames: without sway, with sway (using sway correction). Introduction to Kani's method (rotation contribution method): concept and application to simple frames.
AI Integration: Python automation of moment distribution cycles for any beam/frame configuration. AI-based structural analysis: overview of deep learning models for structural response prediction. Structural optimization using AI: overview of genetic algorithm and topology optimization for frame structures. Introduction to STAAD.Pro and ETABS student versions for structural analysis. Real-world application: AI-assisted analysis of a multi-storey frame – case study using OpenSees. Introduction to Physics-Informed ML for structural mechanics.
Free/Open-Source AI Tools: Python (NumPy, pandas) – Google Colab – Automated moment distribution; OpenSees (free, open-source framework) – Structural analysis validation; SAP2000 v22 student version (free) – Commercial validation; STAAD.Pro CONNECT student edition (free)
– Industry tool exposure; PyNite (free, open-source Python FEM library for 3D frames)

Text Books:

1. Vazirani V.N. and Ratwani M.M., Analysis of Structures – Vol. I & II, Khanna Publications, New Delhi.
2. Reddy C.S., Basic Structural Analysis, Tata McGraw Hill Publishers, 3rd Edition, 2017.
3. Maganti Janardhana K.U. Muthu, Azmi Ibrahim, M. Vijayanand, Basic Structural Analysis,IK International Publishing House.

Reference Books:

1. Punmia B.C., Strength of Materials and Mechanics of Structures, Khanna Publications, New Delhi.
2. Prakash Rao D.S., Structural Analysis, University Press.
3. Kasim Ali A., Structural Analysis, Cengage Publications, 6th Edition, 2020.
4. Vaidyanathan R. and Perumal P., Structural Analysis Vol. I & II, Laxmi Publications, 3rd Edition, 2016.
5. Negi L.S. and Jangid R.S., Structural Analysis, Tata McGraw Hill, 6th Edition.

s

Online Resources:

1. NPTEL: Structural Analysis I – https://archive.nptel.ac.in/courses/105/106/105106050
2. NPTEL: Matrix Methods of Structural Analysis – https://nptel.ac.in/courses/105105108
3. OpenSees Documentation and Tutorials: https://opensees.berkeley.edu
4. SAP2000 Student Version (free): https://www.csiamerica.com/products/sap2000/academic
5. PyNite (free Python FEM): https://github.com/JWock82/PyNite
6. SkyCiv Free Structural Analysis: https://skyciv.com/structural-analysis-software
7. Google Colab for Python matrix analysis: https://colab.research.google.com

Recommended Free & Open-Source AI/Computational Tools:

1. Python (NumPy, SciPy, SymPy, Matplotlib) on Google Colab – Matrix structural analysis, automated moment distribution
2. OpenSees (free, open-source) – Structural analysis framework with Python API
3. PyNite (free, open-source Python library) – 3D FEM structural analysis in Python
4. SAP2000 student version (free) – Industry-standard structural analysis for verification
5. STAAD.Pro CONNECT student edition (free) – Commercial tool exposure and validation
6. scikit-learn (free) on Google Colab – ML for structural response prediction from geometric parameters
7. Sympy (free, open-source) – Symbolic derivation of structural analysis equations

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01208T	PC	Environmental Engineering (AI Integrated)	2	0	2	0	3

Pre-Requisites: Basics of Fluid Mechanics, Concrete Technology, Surveying and Basic Chemistry

Course Objectives:

- To understand sources, characteristics, and quality standards for water supply and wastewater, and their public health significance.
- To understand principles for water intake, conveyance, treatment units, and distribution systems following IS standards.
- To analyze the composition of sewage, sewer systems, and evaluate the performance of wastewater treatment processes.
- To assess the impacts of solid waste and air pollution and apply engineering methods and management strategies for their control.
- To integrate AI, IoT, and computational tools for real-time monitoring, predictive modelling, and intelligent design of environmental engineering systems.

Course Outcomes (COs):

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

CO1: Apply water supply principles, population forecasting methods, design parameters, and IS 10500:2012 quality standards to solve basic problems involving water sources, intakes, and conveyance systems. (L3)

CO2: Apply the principles of unit operations (sedimentation, coagulation, filtration, disinfection, softening) to design water treatment units and distribution pipe networks (Hardy-Cross method), and use AI/ML tools to automate design calculations and predict treatment performance. (L3)

CO3: Analyse sewage quantity and quality parameters including BOD, COD, and DO sag curve (Streeter-Phelps equation), design sewer systems using Manning's equation, and evaluate river self-purification capacity using Python-based computational models. (L4)

CO4: Evaluate the performance of biological wastewater treatment processes (ASP, trickling filters, WSP, septic tanks), design treatment units using standard methods, assess sludge management options, and compare AI-based process control approaches with conventional operational strategies. (L5)

CO5: Design integrated solid waste management systems and air pollution control strategies, implement Gaussian dispersion models and ML-based AQI prediction using Python, and propose AI/IoT-enhanced environmental monitoring and smart waste management frameworks for urban areas. (L6)

CO – PO Articulation Matrix:

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

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

s

Unit I: Water Supply Engineering – Sources, Quantity & Quality:

Hydrological cycle and sources of water: surface water (rivers, lakes, reservoirs) and groundwater (wells, springs, infiltration galleries). Population forecasting methods: arithmetic, geometric, incremental increase, decreasing rate, logistic curve.

Per capita demand: domestic, industrial, commercial, institutional, fire fighting, losses – design period. Quality of water: physical parameters (turbidity, colour, taste, odour, temperature, total solids), chemical parameters (pH, hardness, alkalinity, DO, BOD, COD, chlorides, fluorides, nitrates, iron, arsenic, lead), bacteriological parameters (MPN, coliform test, membrane filter technique). IS 10500:2012 drinking water standards.

Waterborne diseases: typhoid, cholera, dysentery, fluorosis, arsenicosis, methemoglobinemia. Intakes: types (river, reservoir, canal), components.

Conveyance: gravity mains, pumping mains; pipe materials; head loss calculation (Darcy-Weisbach)..

AI Integration: Introduction to AI in Water Supply: overview of AI applications in water infrastructure. ML for population forecasting: applying regression and time-series models (Python/pandas/scikit-learn) to census data. AI for water quality prediction: multi-parameter water quality index (WQI) calculation using Python. IoT-based real-time water quality monitoring: sensor data ingestion and anomaly detection using Python. ML for waterborne disease risk mapping using GIS (QGIS + Python). Overview of smart water meters and AI-driven demand forecasting.

Free/Open-Source AI Tools: Python (pandas, scikit-learn, Matplotlib) – Google Colab (free); QGIS (free, open-source GIS); WHO/BIS water quality databases (free); EPANET (free, USEPA – pipe network analysis)

Unit II: Water Treatment:

Core Content: Unit operations in water treatment: screening, plain sedimentation – overflow rate, detention time, sedimentation efficiency; coagulation and flocculation – jar test, design of coagulation basin; tube settlers.

Filtration: slow sand filter, rapid sand filter (design – hydraulic loading rate, underdrainage, backwashing), pressure filter, dual media filter. Disinfection: chlorination – forms of chlorine, breakpoint chlorination, residual chlorine, chlorine demand, CT concept; UV disinfection; ozone disinfection.

Water Softening: lime-soda process, zeolite (ion exchange) process –.

Defluorination: Nalgonda technique. Iron and arsenic removal.

Desalination: reverse osmosis (RO) concept, nanofiltration – applications.

Water distribution systems: pipe networks, layout types (dead-end, grid iron, ring, radial), design discharge, Introduction to Hardy-Cross method for pipe network analysis. Storage: service reservoirs – elevated tanks, ground-level reservoirs.

AI Integration: AI for jar test optimisation: ML model predicting optimum coagulant dose from turbidity and pH inputs (Python/scikit-learn). Computer vision for floc size analysis (OpenCV). ML-based filter performance prediction: predicting filtered water turbidity from raw water parameters and operational data. Digital twin for water treatment plant: concept and tools. Python automation of Hardy-Cross pipe network analysis (iterative head-balance method). AI for real-time chlorination control using IoT sensor feedback.

Free/Open-Source AI Tools: EPANET 2.2 + EPyT Python wrapper (free, USEPA); Python (NumPy, scikit-learn, OpenCV) – Google Colab (free); WaterGEMS student version; QGIS (free)

Unit III: Sewerage System Design & Wastewater Characteristics:

Core Content : Systems of sewage: separate, combined, partially separate – comparison and selection.

Sewage quantity: dry weather flow (DWF), storm water flow, infiltration, peak flow.

Sewer design: self-cleansing velocity, Manning's equation, storm water drains and sanitary sewers; IS 1742 provisions. Sewer materials and shapes; inverted siphons; manholes – spacing, types.

Sewage characteristics: physical (colour, odour, temperature, solids – suspended, dissolved, volatile), chemical (BOD – 5-day BOD test, BOD reaction rate constant, temperature correction, seeding; COD; nitrogen forms – organic, ammonia, nitrite, nitrate; phosphorus; oil & grease; pH),

biological (coliforms, pathogens, algae). Sampling methods and composite sampling. Significance of BOD, COD, TOC, DO sag curve (Streeter-Phelps equation):, critical DO deficit, critical distance.
AI Integration: Python-based DO sag curve computation using Streeter-Phelps equation (parametric plots using Matplotlib). ML for BOD prediction from easier-to-measure parameters (COD, TSS, temperature) using regression (scikit-learn). AI for sewer flow anomaly detection: statistical process control on continuous SCADA data. Automated sewer network design using Python (Manning's equation automation). Introduction to AI-based inflow and infiltration (I&I) detection in sewer networks using ML and flow meter data.
Free / Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib, scikit-learn) – Google Colab (free); SewerGEMS student version; SWMM (free, USEPA – Storm Water Management Model); QGIS (free)

Unit IV: Wastewater Treatment:
Core Content: Principles of wastewater treatment: preliminary, primary, secondary and tertiary treatment – overview. Preliminary treatment: screens (bar, fine), grit chambers (horizontal flow, aerated) –Primary treatment: sedimentation tanks (rectangular, circular) – surface overflow rate, detention time, weir loading; imhofftanks. Secondary treatment – biological processes: activated sludge process (ASP) – principle, MLSS, SVI, F/M ratio, SRT, aeration methods, aeration tank and secondary clarifier; trickling filters (low rate, high rate) – rotating biological contactors (RBC) – principle. Waste stabilisation ponds (WSP): anaerobic, facultative, maturation ponds . Oxidation ditches. Septic tanks: design as per IS 2470. Sludge management: sources of sludge, thickening (gravity, dissolved air flotation), digestion (anaerobic – mesophilic, thermophilic), dewatering (filter press, centrifuge, drying beds), sludge gas utilisation. Introduction of Tertiary / advanced treatment: nutrient removal (BNR), filtration, UV, activated carbon adsorption. Treated wastewater reuse: irrigation, industrial, groundwater recharge.
AI Integration: ML for activated sludge process control: prediction of effluent BOD and TSS from influent parameters and operational data (Python/scikit-learn Random Forest). AI-based fault detection in wastewater treatment plants using sensor data and anomaly detection (Isolation Forest, scikit-learn). Introduction to digital twins for WWTP: concept using WEST or GPS-X software. Python automation of aeration tank design (ASP) and secondary clarifier sizing. Computer vision for sludge volume index measurement (OpenCV). AI for biogas production optimisation in anaerobic digesters (ML regression).
Free/Open-Source AI Tools: Python (scikit-learn, NumPy, pandas) – Google Colab (free); SWMM (free, USEPA); GPS-X student version; OpenCV (free, open-source); BioWin (trial)

Unit V: Solid Waste Management, Air Pollution & AI Applications:
Core Content: Municipal solid waste (MSW): sources, classification (organic, inorganic, recyclable, hazardous), physical and chemical characteristics; proximate analysis; generation rates. SWM system: collection (primary, secondary), segregation at source, transfer stations, transport. Processing and resource recovery: composting (aerobic windrow, vermi-composting); anaerobic digestion; refuse-derived fuel (RDF); incineration. Sanitary landfill: principles of site selection, design components (liner, leachate collection, gas collection, cover), operation and monitoring; closure and post-closure. Hazardous waste: categories (as per Hazardous Wastes Rules 2016), treatment and disposal. E-waste: types, quantities, formal recycling. Air pollution: classification of pollutants: primary (SO_2, NO_x, CO, SPM, RSPM, VOCs) and secondary (ozone, PAN, photochemical smog); sources – vehicular, industrial, domestic. Effects: health (respiratory diseases, COPD, cancer), vegetation, materials, visibility. NAAQS (National Ambient Air Quality Standards, 2009).
AI Integration: AI for air quality prediction: ML model (Random Forest/LSTM) predicting PM2.5 and AQI from meteorological and traffic data (Python/scikit-learn/TensorFlow on Google Colab). Python implementation of Gaussian plume dispersion model with interactive parameter sensitivity analysis (Matplotlib). ML for solid waste generation prediction: regression model from socio-economic parameters (Python). Computer vision for illegal dumping site detection from satellite/drone imagery (QGIS + deep learning overview). IoT + AI for real-time ambient air quality monitoring network design. AI-based emission source apportionment: receptor modelling using positive matrix factorisation (PMF) – conceptual overview. Google Earth Engine (free) for land-use and green cover change analysis affecting urban air quality.

Free/Open-Source AI Tools: Python (scikit-learn, TensorFlow, pandas, Matplotlib, NumPy) – Google Colab (free); AERMOD (free, US EPA Gaussian dispersion model); QGIS (free); Google Earth Engine (free cloud platform); OpenAQ (free, open-source global air quality data API)

Text Books:

1. Metcalf & Eddy, Wastewater Engineering: Treatment and Resource Recovery, Tata McGraw Hill, 5th Edition (Indian Adaptation), 2020.
2. Peavy H.S., Rowe D.R. and Tchobanoglous G., Environmental Engineering, McGraw Hill, International Edition, 2017.

Reference Books:

1. Punmia B.C. and Jain Ashok Kumar, Water Supply Engineering (Environmental Engineering Vol. I), Laxmi Publications, 17th Edition, 2020.
2. Punmia B.C. and Jain Ashok Kumar, Wastewater Engineering (Environmental Engineering Vol. II), Laxmi Publications, 16th Edition, 2020.
3. Arceivala S.J. and Asolekar S.R., Wastewater Treatment for Pollution Control and Reuse, Tata McGraw Hill, 3rd Edition, 2007.
4. Raju B.S.N., Water Supply and Waste Water Engineering, Tata McGraw Hill, New Delhi.
5. Rao C.S., Environmental Pollution Control Engineering, New Age International, 2nd Edition, 2012.
6. Bhide A.D. and Sundaresan B.B., Solid Waste Management in Developing Countries, INSDOC, New Delhi.
7. IS 10500:2012 – Drinking Water Standards, Bureau of Indian Standards.
8. IS 1742:1983 – Code of Practice for Building Drainage, BIS.
9. IS 2470 Part 1:1985 – Code of Practice for Installation of Septic Tanks, BIS.
10. CPHEEO Manuals (free PDF): Manual on Water Supply and Treatment (3rd Ed.); Manual on Sewerage and Sewage Treatment (3rd Ed.), Ministry of Housing & Urban Affairs, Govt. of India.

Online Resources:

1. NPTEL: Water and Waste Water Engineering – https://nptel.ac.in/courses/105105172
2. NPTEL: Environmental Engineering – https://nptel.ac.in/courses/105101210
3. NPTEL: Air Pollution and Control – https://nptel.ac.in/courses/105105013
4. CPHEEO Manuals (free download): https://cpheeo.gov.in
5. USEPA SWMM (free storm water model): https://www.epa.gov/water-research/storm-water-management-model-swmm
6. EPANET 2.2 (free water distribution model): https://www.epa.gov/water-research/epanet
7. AERMOD View / AERMOD (free US EPA dispersion model): https://www.epa.gov/scram
8. OpenAQ (free global air quality data): https://openaq.org
9. Google Earth Engine (free cloud geospatial): https://earthengine.google.com
10. Virtual Labs – Environmental Engineering (IIT): https://eerc-iitb.vlabs.ac.in

Recommended Free & Open-Source AI/Computational Tools:

1. Python (pandas, NumPy, SciPy, Matplotlib, scikit-learn) on Google Colab – Water quality index, DO sag, BOD prediction, ASP design automation, AQI ML modelling
2. TensorFlow / Keras (free, open-source) on Google Colab – LSTM for AQI/BOD time-series prediction; CNN for illegal dumping detection
3. EPANET 2.2 + EPyT Python wrapper (free, USEPA) – Pipe network Hardy-Cross automation and water distribution modelling
4. SWMM (free, USEPA) – Storm water and sewer network simulation

5. AERMOD (free, USEPA) – Gaussian air dispersion modelling for industrial stacks
6. QGIS (free, open-source) – Spatial mapping of water quality, solid waste sites, air quality monitoring stations
7. Google Earth Engine (free cloud) – Satellite-based land-use analysis affecting water and air quality
8. OpenAQ API (free, open-source) – Real-time and historical air quality data for ML model training
9. OpenCV (free, open-source) on Google Colab – Floc image analysis, illegal dump detection from imagery

Practical/Lab Component (Pr-2 hrs/week):

The following experiments / exercises shall be performed concurrently with theory:

1. Turbidity and Colour Test: Determine turbidity (NTU) and apparent/true colour of given water samples using nephelo-turbidimeter and Nessler's tubes; compare with IS 10500:2012 limits.
2. Jar Test for Coagulation: Determine optimum coagulant (alum) dose and pH for a turbid water sample; plot turbidity vs. dose curve; compare with ML-predicted optimum dose using Python regression model (Google Colab).
3. Rapid Sand Filter Design Problem: Given raw water turbidity and design flow, compute filter dimensions, under drain spacing, and backwash velocity; verify output using Python automation script.
4. Chlorine Demand and Residual: Determine chlorine demand of a water sample by preparing breakpoint chlorination curve; estimate residual chlorine using ortho-tolidine method.
5. 5-Day BOD Test: Prepare BOD dilution bottles for given wastewater sample; measure DO on day 0 and day 5 using Winkler's method or DO meter; calculate BOD5; determine BOD rate constant k using Thomas graphical method.
6. COD Determination: Determine COD of wastewater sample using open reflux dichromate method; compare BOD: COD ratio to assess biodegradability.
7. DO Sag Curve – Python Lab: Using given river discharge, re-aeration and de-oxygenation rate constants and initial deficit, compute and plot DO sag curve using Python (SciPy ODE solver); determine critical DO and critical distance.
8. Settling Velocity Test: Determine settling velocity of suspended solids using a standard column test (batch settling experiment); plot settling velocity distribution curve; design settling tank using overflow rate.
9. Hardy-Cross Pipe Network Analysis – Python Lab: Solve a given 6-pipe looped network by hand (one iteration) and then automate using Python (EPyT/EPANET) to compare results and visualise pressure distribution.
10. Air Quality Monitoring Study: Using available air quality data (OpenAQ API or CPCB station data), download PM2.5 and AQI data for a given city; perform data analysis (trend, seasonality) using Python/pandas; build a simple Random Forest AQI predictor.
11. Gaussian Plume Model – Python Lab: Implement Holland's effective stack height formula and Gaussian concentration equation in Python; plot ground-level concentration vs. downwind distance for stable and unstable atmospheric conditions; perform sensitivity analysis on stack height and emission rate.
12. Solid Waste Characterisation: Determine proximate analysis (moisture content, volatile matter, ash, fixed carbon) of a mixed MSW sample; compute calorific value and assess suitability for RDF/incineration.

Sample AI Integration Mini-Project Ideas for Students

- Smart Water Quality Monitoring Dashboard (Python + Google Colab): Ingest real-time data from an OpenAQ-style API or simulated IoT sensor stream; compute a multi-parameter Water Quality Index (WQI) and raise colour-coded alerts; visualise trends using Matplotlib/Plotly.
- AI-Based BOD Prediction from Physicochemical Parameters (Python/scikit-learn): Collect or download a wastewater quality dataset (Kaggle/UCI); train and compare Linear Regression, Random Forest, and Gradient Boosting models; evaluate with RMSE and R^2.
- DO Sag Curve Sensitivity Dashboard (Python): Build an interactive Matplotlib/ipywidgets tool that plots the Streeter-Phelps DO sag curve in real time as users slide parameters (k_1, k_2, L_0, D_0, U); illustrate how river self-purification varies with season and effluent load.
- Automated Aeration Tank Design (Python): Implement the complete McKinney / IS-based activated sludge process design procedure (SRT, F/M, MLSS, aeration volume, oxygen demand, secondary clarifier) as a Python function; validate against a textbook example and generate a formatted design report.
- AQI Prediction and Mapping for a City (Python + QGIS): Download CPCB/OpenAQ PM2.5 and NO_2 historical data; train an LSTM model for next-day AQI prediction; plot a spatial AQI heat map using QGIS + Python geopandas.
- Gaussian Plume Web Tool (Python/Streamlit – free): Develop a Streamlit web app where users input stack parameters and atmospheric stability class; the app computes and visualises ground-level concentration isopleths in real time.

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01205P	PC	Fluid Mechanics Lab	0	0	0	3	1.5

Pre-Requisites: Basics of Engineering Physics, Engineering Mathematics and Fluid Mechanics

Course Objectives:

- To verify Bernoulli's theorem and determine discharge coefficients for orifice, Venturimeter and Orifice meter.
- To calibrate triangular, rectangular and trapezoidal weirs and determine head–discharge relationships.
- To determine minor losses and friction factor in pipe flow.
- To analyze energy loss and flow characteristics in a hydraulic jump.
- To automate experimental data analysis and develop AI-assisted flow prediction models using Python and scikit-learn.

Course Outcomes (COs):

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

CO1: Verify Bernoulli's theorem experimentally and determine the coefficient of discharge (Cd) for a small orifice by the constant head method, Venturimeter and Orifice meter; record piezometric heads, measure actual discharge by volumetric method and compute Cd for each device. (L3)

CO2: Determine head–discharge relationships and calibration constants for triangular notch, rectangular notch and trapezoidal weir; plot Q vs. H curves on log-log scale; verify standard discharge formulae; compare Cd values across notch and weir types and analyse variation with head. (L4)

CO3: Determine minor (local) loss coefficients K for pipe fittings and friction factor f for pipe flow using the Darcy-Weisbach equation; plot f vs. Re on the Moody diagram region; analyse energy loss in a hydraulic jump by computing upstream and downstream depths, Froude number and specific energy. (L4)

CO4: Analyse energy dissipation and flow characteristics during hydraulic jump formation; compute energy loss, efficiency and conjugate depth ratio; evaluate experimental results against Belanger's equation and compare flow measurement accuracy across Venturimeter, orifice meter and notches. (L5)

CO5: Automate Cd computation and Q vs. H plotting using Python (Matplotlib); develop ML-based discharge prediction for weirs using scikit-learn Polynomial Regression; perform hydraulic jump energy analysis using Python (NumPy); evaluate model accuracy using R^2 and RMSE and compare with standard formulae. (L6)

CO – PO Articulation Matrix:

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

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

List of Field Work/Lab Experiments:

Part A- Regular Experiments:

1. Experimental Verification of Bernoulli's Theorem
2. Determination of Coefficient of Discharge of a Small Orifice by using the constant head method
3. Determination of the discharge coefficient and flow measurement characteristics of a Venturimeter
4. Evaluation of flow measurement accuracy and determination of the discharge coefficient of an Orifice Meter
5. Determination of the relationship between head and discharge for a triangular notch
6. Determination of discharge characteristics and calibration constants of a rectangular notch
7. Determination of discharge characteristics and calibration constants of a Trapezoidal Weir
8. Determination of Minor Losses in Pipe Flow
9. Determination of Friction Factor in Pipe Flow
10. Determination of Energy Loss in a Hydraulic Jump
11. Analysis of energy dissipation and flow characteristics during hydraulic jump formation

Part B- AI Integration Lab Exercises:

1. Discharge Coefficient Automation (Python / Google Colab): input measured heads and discharge data from Experiments 2, 3 and 4; automate Cd computation for each run; plot Q_{0ns} vs. Q_{the}^{or} and Cd vs. head curves using Matplotlib; compute mean Cd and percentage deviation from standard IS values.
2. ML-Based Flow Prediction (scikit-learn): using experimental notch data from Experiments 5–7 (head H, Cd, temperature), train a Polynomial Regression model to predict discharge Q for unseen heads; compare ML prediction with standard notch formula; evaluate R^2 and RMSE; discuss model validity range.
3. Hydraulic Jump Analysis (Python / NumPy): input upstream depth y_1, downstream depth y_2 and discharge Q from Experiments 10 and 11; compute Fr_1, theoretical y_2 (Belanger equation), energy loss ΔE and efficiency; compare with measured values; plot energy dissipation vs. upstream Froude number.

AI Integration:

AI integration adds three exercises following the regular experiments: (i) Python automation of Cd computation (Experiments 2–4) – processing head and discharge data, plotting actual vs. theoretical Q and Cd vs. H, eliminating manual tabulation; (ii) scikit-learn Polynomial Regression for weir flow prediction (Experiments 5–7) – training on measured H-Q data and predicting Q for unseen heads, evaluated against the standard notch formula; (iii) Python hydraulic jump analysis (Experiments 10–11) – computing Froude number, Belanger's depth ratio, energy loss and efficiency, comparing with measured values and plotting energy dissipation curves – reinforcing the connection between physical observation and theoretical hydraulics.

Free / Open-Source AI Tools: Python (NumPy, Matplotlib, scikit-learn) – Google Colab (free); EPANET 2.2 + EPyT (free, USEPA); Virtual FM Lab – NPTEL / VLabs (free).

Recommended Free & Open-Source AI / Computational Tools Tools :

- Python (NumPy, Matplotlib, scikit-learn) on Google Colab (free) – Cd automation, Q vs. H plotting, ML-based weir flow prediction, hydraulic jump energy analysis
- EPANET 2.2 (free, USEPA) + EPyT Python wrapper (free) – pipe network head-loss computation and visualisation

- Virtual Fluid Mechanics Lab – NPTEL / VLabs (free): https://vlabs.ac.in – interactive Bernoulli, notch and Reynolds simulations
- NIST WebBook (free) – fluid properties (density, viscosity) at experimental temperatures
- Scilab (free, open-source) – alternative numerical tool for friction factor and hydraulic jump computations

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26PC01206P	PC	Concrete Technology Lab	0	0	0	3	1.5

Pre-Requisites: Basics of Engineering Chemistry (basic materials knowledge); concurrent: Concrete Technology

Course Objectives:

- To conduct standard IS tests on cement to determine physical and strength properties relevant to civil engineering applications.
- To evaluate fine and coarse aggregate properties including grading, fineness modulus, specific gravity, water absorption and bulking.
- To measure and compare workability of fresh concrete using compaction factor, slump and Vee-bee test methods.
- To test hardened concrete for mechanical properties (compressive strength, modulus of rupture, Young's modulus, Poisson's ratio, split tensile strength) and demonstrate non-destructive testing.
- To automate aggregate grading data processing and mix design using Python and develop ML-based compressive strength prediction using scikit-learn.

Course Outcomes (COs):

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

CO1: Conduct standard IS tests on cement – normal consistency, fineness (IS 4031 Part 1), initial and final setting time (IS 4031 Part 5), specific gravity, soundness (Le-Chatelier, IS 4031 Part 3) and compressive strength of cement mortar cubes (IS 4031 Part 6) – using appropriate apparatus; record observations and compute results correctly. (L3)

CO2: Perform sieve analysis on fine and coarse aggregates; determine fineness modulus and grading zone; measure specific gravity of fine and coarse aggregates (IS 2386 Part 3); determine water absorption of coarse aggregates and water absorption and bulking of sand; plot and interpret grading curves against IS 383 zone limits. (L4)

CO3: Measure the workability of fresh concrete using the compaction factor test, slump test and Vee-bee test (IS 1199); compare workability values obtained by the three methods for identical mix proportions; analyse the influence of water-cement ratio and admixtures on workability. (L4)

CO4: Test hardened concrete for compressive strength and modulus of rupture (IS 516), Young's modulus and Poisson's ratio, and split tensile strength (IS 5816); evaluate concrete quality and compare results with mix design targets; demonstrate and interpret non-destructive testing (rebound hammer / UPV) results on concrete specimens. (L5)

CO5: Automate sieve analysis data processing and grading curve plotting using Python (Matplotlib) with IS 383 zone overlay; develop a Random Forest ML model (scikit-learn) for 28-day compressive strength prediction from mix parameters; implement IS 10262:2019 mix design as a Python function and verify against a worked example. (L6)

CO – PO Articulation Matrix:

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

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

List of Field Work/Lab Experiments:

Part A- Regular Experiments:

1. Tests on Cement
    - Normal Consistency and Fineness of cement
    - Initial setting time and Final setting time of cement
    - Specific gravity and soundness of cement
    - Compressive strength of cement
2. Tests on Fine Aggregates
    - Grading and fineness modulus of Fine aggregate by sieve analysis
    - Specific gravity of fine aggregate
    - Water absorption and Bulking of sand
3. Tests on Coarse Aggregates
    - Grading of Coarse aggregate by sieve analysis
    - Specific gravity of coarse aggregate
    - Water absorption of Coarse aggregates
4. Tests on Fresh Concrete
    - Workability of concrete by compaction factor method
    - Workability of concrete by slump test
    - Workability of concrete by Vee-bee test
5. Tests on Hardened Concrete
    - Compressive strength of cement concrete and Modulus of rupture
    - Young's Modulus and Poisson's Ratio
    - Split tensile strength of concrete
    - Non-Destructive testing on concrete (for demonstration)

Part B- AI Integration Lab Exercises:

1. Sieve Analysis Data Automation (Python / Google Colab): input sieve analysis data for fine and coarse aggregates from Experiments 2.1 and 3.1; automate fineness modulus computation; plot grading curves for both aggregates using Matplotlib and overlay the IS 383 permissible grading zone limits; flag whether the aggregate falls within the specified grading zone.
2. ML-Based Compressive Strength Prediction (Python / scikit-learn): using a provided dataset of mix parameters (w/c ratio, cement content, fine aggregate content, coarse aggregate content, admixture dosage, curing age), train a Random Forest Regression model to predict 28-day compressive strength; compare predictions with experimental values from Experiment 5.1; evaluate model accuracy using R^2 and RMSE.
3. IS 10262:2019 Mix Design Automation (Python): implement the complete IS 10262:2019 concrete mix design procedure as a Python function accepting target mean strength (f'ck), exposure condition, aggregate type and maximum size as inputs; output the design w/c ratio, cement content, water content and aggregate proportions; verify the design output against a manually worked mix design example.

AI Integration:

AI integration adds three exercises following the regular experiments: (i) Python sieve analysis automation – processing grading data for fine and coarse aggregates from Experiments 2.1 and 3.1, automating fineness modulus computation and plotting grading curves overlaid with IS 383 zone limits using Matplotlib; (ii) scikit-learn Random Forest regression for 28-day compressive strength prediction – training on a mix parameter dataset and comparing predicted values with Experiment 5.1 results, evaluated with R^2 and RMSE; (iii) Python implementation of IS 10262:2019 mix design procedure as a function – accepting target strength, exposure condition and aggregate data and returning full mix proportions, verified against a manually worked example – demonstrating how routine design calculations can be automated for AI-assisted concrete technology practice.

Free / Open-Source AI Tools: Python (NumPy, Pandas, Matplotlib, scikit-learn) – Google Colab (free); Virtual CT Lab – VLabs NIT Karnataka (free); LibreOffice Calc (free).

Recommended Free & Open-Source AI / Computational Tools Tools :

- Python (NumPy, Pandas, Matplotlib, scikit-learn) on Google Colab (free) – sieve analysis automation, grading curve plotting with IS 383 overlay, ML strength prediction, IS 10262 mix design automation
- scikit-learn (free, open-source) – Random Forest Regression for 28-day compressive strength prediction from mix design parameters
- Virtual Concrete Technology Lab – VLabs NIT Karnataka (free): https://ced-nitk.vlabs.ac.in – interactive simulation of slump, compaction factor, compressive strength and NDT experiments
- IS 10262:2019 (BIS, free institutional access) – reference for mix design automation exercise
- IS 383:2016 and IS 2386 (BIS) – reference for grading zone overlay and aggregate testing standards
- LibreOffice Calc (free, open-source) – standardised data recording templates for all cement, aggregate and concrete tests

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26SE01202S	PC	Auto CAD for Civil Engineers	1	0	2	0	2

Pre-Requisites: Basics of Engineering Graphics, Building Planning and Drawing and Basic computer skills.

Course Objectives:

- To develop proficiency in the AutoCAD interface, command line, layer management and basic 2D drawing commands for civil engineering applications.
- To apply 2D editing commands and IS 962 dimensioning conventions to create accurate, annotated civil engineering drawings.
- To prepare complete standard civil engineering drawings – floor plan, elevation, section, reinforcement detail and road drawings – using AutoCAD.
- To use advanced AutoCAD features including blocks, attributes, XREFs, layouts and Auto LISP scripting for professional, automated drawing production.
- To create 3D solid models of civil structures, export to BIM (IFC), and apply AI-assisted tools for quantity take-off and generative design.

Course Outcomes (COs):

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

CO1: Apply AutoCAD interface, drawing setup, basic 2D commands and drafting aids to create accurate engineering drawings. (L3)

CO2: Apply 2D editing, dimensioning and annotation commands to modify and document engineering drawings as per IS 962. (L3)

CO3: Prepare civil engineering drawings such as plans, elevations, sections, site plans, reinforcement details and road profiles as per IS 962 and NBC 2016. (L4)

CO4: Apply advanced AutoCAD tools, blocks, references, layouts and automation techniques for efficient professional drafting and drawing management. (L5)

CO5: Create 3D civil engineering models, integrate BIM and quantity take-off tools, and apply Python and AI-assisted generative design for civil engineering applications. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: AutoCAD Interface, Layer Management and Basic 2D Drawing Commands:

Core Content: Introduction to CAD and AutoCAD Interface: Concept and advantages of Computer-Aided Drafting (CAD) over manual drafting; applications in civil engineering. AutoCAD workspace: Ribbon, toolbars, command line, Dynamic Input, Status bar, Model space and Paper space. Setting up a drawing: drawing units (IS conventions), drawing limits, grid and snap settings. Layer management: creating layers, assigning colour, linetype and lineweight as per IS 962; current layer, layer visibility, freeze and lock.

Basic 2D Drawing Commands: Line, Circle (3 methods), Arc (5 methods), Rectangle, Polygon, Ellipse, Spline, Polyline, Multiline. Point, Donut, Hatch (pattern, solid, gradient). Coordinate entry: absolute, relative Cartesian and polar (@r<θ). Object Snaps (OSNAP): Endpoint, Midpoint, Center, Intersection, Perpendicular, Tangent, Nearest. Running OSNAP, Object Snap Tracking, Ortho mode, Polar Tracking. Zoom and Pan: wheel zoom, window zoom, extents, realtime.

AI Integration: Introduction to AI in CAD: overview of how AI is transforming drafting workflows. AutoCAD AutoComplete and Auto-correct on the command line – how ML suggestions accelerate command entry. Autodesk AI (Autodesk Forma integration concept). Introduction to using ChatGPT / AI assistants for looking up AutoCAD commands and troubleshooting drawing errors interactively. AI-assisted layer naming conventions: how BIM software automates layer management using predefined IS/ISO standards. Google Colab Python tutorial: drawing simple geometric shapes programmatically using the ezdxf library (free, open-source) – introduction to how code can generate DXF/DWG content automatically.

Free / Open-Source Tools: AutoCAD student version (free – autodesk.com/education); LibreCAD (free, open-source 2D CAD); ezdxf Python library (free) – Google Colab; FreeCAD (free, open-source 3D CAD).

Unit II: 2D Editing Commands, Dimensioning and Annotation:

Core Content: 2D Editing Commands: Erase, Undo, Redo; Move, Copy, Rotate (with base point and reference angle); Mirror (with and without deleting source); Offset (distance and through point); Trim, Extend (with Shift key toggle); Fillet (radius, polyline option), Chamfer (distance and angle methods); Stretch, Scale, Lengthen; Array – Rectangular array, Polar array, Path array. Grip editing. Properties panel and Quick Properties. Match Properties.

Dimensioning and Annotation: Dimension styles: creating and modifying (text height, arrowhead size, tolerances) as per IS 962. Dimension types: Linear, Aligned, Angular, Radial, Diameter, Arc Length, Ordinate, Baseline, Continue. Quick Dimension; Dimension Override; Editing dimensions using DDEDIT and grips. Text: Single Line Text (DTEXT), Multiline Text (MTEXT) – text styles, font, height, oblique angle. Field text. Tables: inserting and formatting data tables in drawings. Leaders and Multileaders for annotations.

AI Integration: AI in Drawing Editing and Annotation: Autodesk AutoCAD Smart Blocks – how AI recognises block geometry and suggests matching blocks from the drawing; introduction to this feature in AutoCAD 2024+. AI-based drawing error detection: overview of tools that automatically flag unclosed polylines, duplicate lines, overlapping objects and IS dimension standard violations. Automated title block population using Field text linked to drawing properties (non-AI but digital automation). Introduction to Python ezdxf for automated dimension generation: writing a Python script to add linear dimensions to a set of objects programmatically from a coordinate list, reducing repetitive manual work.

Free / Open-Source Tools: AutoCAD student version (free); LibreCAD (free); ezdxf Python library – Google Colab (free); DraftSight free version.

Unit III: Civil Engineering Drawings – Plans, Sections, Reinforcement and Road Drawings:

Core Content: Civil Engineering Plan and Section Drawings: Setting up a drawing template for civil engineering: IS 962 conventions for line types (centre line, hidden line, cut line, break line), text heights and border formats. Drawing a site plan: north arrow, scale bar, boundary lines, setback lines, access road, open spaces. Floor plan of a single-storey residential building: walls (to scale), doors and windows (IS symbols), columns, stairs, toilet. Elevation drawing: fenestration, sunshade, parapet, floor levels. Section drawing: footing, plinth, floor, roof slab with hatching for materials (brick, concrete, soil).

AI Integration: AI-Assisted Civil Engineering Drawing Production: Introduction to AutoCAD Architecture (student version free) – how it automates wall, door and window placement using parametric objects. AI-based drawing sheet checking: overview of tools that verify drawing against NBC 2016 setback and FAR requirements automatically. Python ezdxf for automated floor plan export: reading a room layout from a CSV file and generating AutoCAD DXF floor plan geometry programmatically – students run a provided script, inspect the output in LibreCAD, and modify room dimensions.

Free / Open-Source Tools: AutoCAD Architecture student version (free); AutoCAD Civil 3D student version (free); Python ezdxf – Google Colab (free); LibreCAD (free); QGIS (free) – for site plan base map.

Unit IV: Structural and Road Engineering Drawings:

Core Content: Structural and Road Engineering Drawings: Reinforcement drawing of an isolated footing: plan and section with rebar schedule. Column reinforcement detail: rectangular ties, spiral ties. Simply supported beam reinforcement: main bars, stirrups, cranked bars. Road cross-section drawing: carriageway, shoulders, camber, embankment slope, drain. Road longitudinal profile (L-section): reduced levels, formation levels, earthwork, vertical curves. Contour plan: plotting spot levels and interpolating contours.

AI Integration: Introduction to road design automation: Civil 3D (AutoCAD-based, student version free) for automated L-section and cross-section generation from alignment and profile data.

Free / Open-Source Tools: AutoCAD Architecture student version (free); AutoCAD Civil 3D student version (free); Python ezdxf – Google Colab (free); LibreCAD (free); QGIS (free) – for site plan base map.

Unit V: Layouts, Plot Settings and Drawing Output:

Core Content: Layouts, Plot Settings and Drawing Output: Model space vs. Paper space layouts. Creating and managing layout tabs. Viewports: creating, locking, scaling (1:50, 1:100 etc. as per IS 962). Annotative objects and annotative scale. Page setup manager: paper size, plot device, plot style (CTB). Plot to PDF: creating professional drawing output with border and title block. Sheet Set Manager: organising multiple drawing sheets for a project. Transmittal: packaging DWG files with XREFs for submission.

Free/Open-Source AI Tools: Free / Open-Source Tools: AutoCAD Architecture student version (free); AutoCAD Civil 3D student version (free); Python ezdxf – Google Colab (free); LibreCAD (free); QGIS (free) – for site plan base map

List of Practical Exercises (P – 2 hrs / week)

The following exercises are to be performed in the AutoCAD computer lab (or on personal laptops using the free AutoCAD student version). Each exercise corresponds to the unit covered in the concurrent lecture session.

Unit I – Interface, Layers and Basic Drawing

1. Setting up an AutoCAD drawing file: configure units (millimetres), drawing limits (A1 sheet: 841×594 mm), grid and snap; create a standard layer table with IS 962 line types and colours for a civil engineering drawing.
2. Basic 2D drawing: draw a given geometric shape (rectangle with inscribed circle, hexagon, arc and polyline) using absolute, relative Cartesian and polar coordinate entry; apply OSNAP modes.
3. Drawing a standard A1 drawing sheet border and title block with all required fields using Lines, Rectangles and Single Line Text on appropriate layers; save as a drawing template (.dwt).

Unit II – Editing and Dimensioning

1. Editing exercise: given a partially drawn civil engineering detail, complete it using Move, Copy, Rotate, Mirror, Offset, Trim, Fillet and Chamfer; reproduce the full detail matching all given dimensions.
2. Array exercise: create a rectangular array of columns (150×150 mm) on a 4×3 grid with 4 m spacing; apply polar array to create a circular tank with equally spaced pillars.
3. Dimensioning and annotation: add a complete IS 962 compliant dimension scheme to a given building plan – overall dimensions, room dimensions, wall thickness callouts and opening widths; add a notes table using MTEXT.

Unit III – Civil Engineering Drawings

1. Floor plan drawing: draw the floor plan of a given 2-bedroom residential building (from dimensions provided) to a scale of 1:50; include walls, doors, windows (IS symbols), stairs, bathroom and kitchen; add all dimensions and north arrow.
2. Elevation and section drawing: draw the front elevation and longitudinal section (AA) of the building from Exercise 1; apply appropriate hatching for materials (brick, concrete, soil, glass); add level annotations.

3. Reinforcement drawing: draw the plan and cross-section of an isolated footing (given size, depth and rebar schedule); produce a reinforcement bar bending schedule table linked to the drawing.
4. Road drawings: draw the typical cross-section of a two-lane rural road (IRC: 73) showing carriageway, shoulders, camber, side slopes, drain and embankment; draw the longitudinal profile (L-section) for a 500 m stretch with given reduced levels.

Unit IV – Blocks, XREFs and Layouts

1. Block creation: create standard civil engineering symbol blocks (north arrow, section cut symbol, column circle, drainage inlet) with attributes (block number, description, date); insert into a drawing and extract attribute data to a CSV table.
2. Dynamic block: create a dynamic door block with a linear parameter for width (range 600–1200 mm in 100 mm steps) and a flip parameter for door swing direction; insert and test the dynamic behaviour.
3. XRef exercise: attach an architectural plan DWG as an XRef; overlay the structural grid and column layout as a separate DWG; bind the XRef for final submission; demonstrate layer management across XRef files.
4. Layouts and plotting: set up two paper space layouts (1:50 floor plan on A1; 1:20 section detail on A3); add annotative dimensions; configure CTB plot styles; export both sheets as PDF.

Unit V – 3D Modelling and AI Integration

1. 3D solid modelling – structural components: model a 230×300 mm column (4 m height), a 230×450 mm beam (6 m span) and a 150 mm flat slab (5×6 m) as 3D solids; assemble a complete bay of a framed structure using Union; extract a section view using Section Plane.
2. 3D staircase modelling: model a dog-legged staircase (given riser and tread dimensions, flight width) as a 3D solid; produce Flatshot plan and section views from the 3D model; compare with hand-drawn section.
3. Quantity take-off using Python ezdxf (AI Lab): run a provided Python script (Google Colab) that reads a DXF file with 3D solid objects, extracts object volumes using MASSPROP data, and outputs a material quantity table in Pandas; modify the script to add surface area computation.
4. IFC export and BIM viewing: export the framed structure 3D model as IFC from FreeCAD (free); open in BlenderBIM (free) to view the BIM model; verify that structural elements are correctly classified; demonstrate in Autodesk Viewer (free cloud).

Text Books:

1. Sham Tickoo, "AutoCAD 2024: A Problem-Solving Approach (Basic and Intermediate)", CADCIM Technologies, 2024.
2. George Omura and Brian C. Benton, "Mastering AutoCAD 2024 and AutoCAD LT 2024", John Wiley & Sons (Sybex), 2023.
3. Subramaniam P., "AutoCAD for Civil Engineers", New Age International Publishers, 2020.
4. IS 962:1989, "Code of Practice for Architectural and Building Drawings", Bureau of Indian Standards, New Delhi.

Reference Books:

1. David Cohn, "Inside AutoCAD 2024", Peachpit Press, 2023.
2. Ellen Finkelstein, "AutoCAD 2024 and AutoCAD LT 2024 Bible", John Wiley & Sons, 2023.
3. Saeed Moaveni, "Engineering Fundamentals: An Introduction to Engineering" (Chapter on CAD), Cengage Learning, 2011.

4. Autodesk, "AutoCAD 2024 User's Guide" (free PDF from Autodesk Knowledge Network).
5. NBC 2016 – National Building Code of India, Bureau of Indian Standards (for drawing standards and building bye-laws).
6. IRC: 35–1997, "Code of Practice for Road Markings"; IRC: 73–1980, "Geometric Design Standards for Rural (Non-Urban) Highways", Indian Roads Congress.

Online Resources:

1. Autodesk Education Community (free AutoCAD student version): https://www.autodesk.com/education/edu-software
2. Autodesk AutoCAD Official Tutorials – https://help.autodesk.com/view/ACD/2024/ENU
3. Autodesk Fusion 360 (free for education): https://www.autodesk.com/education/edu-software
4. ezdxf Python Library (free, open-source DXF/DWG): https://ezdxf.readthedocs.io
5. LibreCAD (free, open-source 2D CAD): https://librecad.org
6. FreeCAD with BIM Workbench (free, open-source): https://freecadweb.org
7. BlenderBIM (free, open-source BIM / IFC): https://blenderbim.org
8. NPTEL: Computer-Aided Design – https://nptel.ac.in/courses/112102033
9. Autodesk Viewer (free online DWG / IFC viewer): https://viewer.autodesk.com
10. Google Colab for Python AutoCAD automation: https://colab.research.google.com

Recommended Free & Open-Source AI/Computational Tools:

1. AutoCAD 2024 student version (free – autodesk.com/education) – primary drafting tool; includes AI AutoComplete, Smart Blocks and AutoLISP
2. AutoCAD Civil 3D student version (free – autodesk.com/education) – automated road L-section and cross-section generation from alignment and profile data
3. Autodesk Fusion 360 (free for education) – AI-based generative design exploration for structural components
4. LibreCAD (free, open-source) – alternative to AutoCAD for 2D drafting (librecad.org)
5. FreeCAD with BIM Workbench (free, open-source) – DWG to IFC conversion for BIM export (freecadweb.org)
6. BlenderBIM (free, open-source) – IFC model creation and viewing; interoperable with FreeCAD and AutoCAD exports
7. ezdxf Python library (free, open-source) on Google Colab – programmatic DXF creation, 3D solid property extraction, automated title block population
8. Autodesk Navisworks student version (free) – multi-discipline XRef coordination and AI-assisted clash detection
9. Autodesk Viewer (free cloud) – online DWG / IFC viewing without installing software

B. Tech. – II Year II Semester

Course Code	Category	Name of the Course	L	T	Pr	P	C
26HMHS206A	MC	**Technical Paper Writing & IPR** (Common to all branches of Engineering)	2	0	0	0	0

Pre-Requisites: Communicative English.

Course Objectives:

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

Course Outcomes (COs):

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

CO1: Apply the principles, types, and structure of technical reports and the fundamentals of Intellectual Property Rights (IPR) to prepare and organize basic technical documents. (L3)

CO2: Apply the principles of technical writing to produce well-structured laboratory reports, project reports, and technical proposals. (L3)

CO3: Analyse technical documents for clarity, coherence, accuracy, and compliance with formatting standards using AI-powered review tools. (4)

CO4: Evaluate patent documents, prior art, and IP claims to assess novelty and inventiveness of an engineering innovation. (L5)

CO5: Create a complete technical report and draft a patent disclosure for an engineering project, integrating AI tools for writing assistance. (L6)

CO – PO Articulation Matrix:

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

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

Unit I: Foundations of Technical Communication:

Core Concepts: 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: AI Tool: Grammarly (free) – analyse a technical paragraph for passive voice overuse and nominalization, **AI Tool:** Hemingway Editor – paste a technical section and improve readability score to Grade 10 or below, Activity: Compare two versions of a technical document (before and after AI editing) and document changes

Unit II: Technical Report Writing:

Core Concepts: 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 Tool: Elicit.org (free) – AI-powered literature search; find and summarise 5 relevant papers, AI Tool: Zotero (open-source reference manager) – manage citations and auto-generate bibliographies, AI Tool: ChatGPT – generate a first-draft abstract; critique and rewrite to meet technical writing standards, Plagiarism Check: Use Turnitin (institutional access) or Copyleaks (free tier) to verify originality

Unit III: Research Proposals and Scientific Writing:

Core Concepts: 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: AI Tool: Semantic Scholar (free) – search for open-access papers in your domain; analyse IMRAD structure, AI Tool: Writefull (free academic writing app) – get AI feedback on scientific language in a draft methods section, Activity: Write a 1-page research proposal for a small project; use AI for structural review and language polishing

Unit IV: Intellectual Property Rights – Fundamentals:

Core Concepts: 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 (Non-obviousness), Industrial Applicability, Patent Search: Prior art search using databases, Copyright in Engineering: Software copyright, Open-source licenses (GPL, MIT, Apache)

AI Integration: AI Tool: Google Patents (free) – conduct a prior art search on a given invention idea, AI Tool: Lens.org (free, open-source patent database) – search and analyse patent documents, Activity: Identify 3 patents related to a chosen engineering domain and analyse their claims section, Discussion: How does AI help in patent drafting? Use AI tools like PatSnap or Examiner (where accessible)

Unit V: Patent Drafting and IP Strategy:

Core Concepts: 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: AI Tool: ChatGPT – draft a set of patent claims for a simple mechanical invention; critique for clarity and scope, AI Tool: IP India Online Portal (ipindia.gov.in) – navigate and understand the e-filing system, Capstone Project: Write a complete patent disclosure document for a project-based invention idea, Activity: Use AI to convert a project report executive summary into a patent abstract; compare the two

Text Books:

1. T1. Markel, Mike & Selber, Stuart. (2022). Technical Communication (13th ed.). Bedford/St. Martin's, New York.
2. T2. Ahuja, B.N. & Ahuja, S.S. (2021). Intellectual Property Rights: Patents, Trade Marks, Copyrights and Related Topics (4th ed.). Khanna Publishers, New Delhi.

Reference Books:

1. R1. Lannon, John & Gurak, Laura. (2020). Technical Communication (15th ed.). Pearson Education.
2. R2. Anderson, Paul V. (2019). Technical Communication: A Reader-Centred Approach (9th ed.). Cengage.
3. R3. Meganathan, R. (2019). Intellectual Property Rights for Engineers. Universities Press (India).
4. R4. Cornish, William R. & Llewelyn, David. (2019). Intellectual Property: Patents, Copyright, Trademarks and Allied Rights (9th ed.). Sweet & Maxwell.
5. R5. Eisenberg, Anne. (2018). Effective Technical Communication. Tata McGraw-Hill.
6. R6. World Intellectual Property Organization (WIPO). (2022). WIPO Intellectual Property Handbook. WIPO Geneva.

Online Resources:

1. IP India – Patent Office: www.ipindia.gov.in
2. Google Patents: patents.google.com
3. Lens.org – Free Patent and Scholar Search: www.lens.org
4. WIPO – Patent Resources: www.wipo.int
5. Purdue OWL – Technical Writing: owl.purdue.edu/owl/subject_specific_writing/professional_technical_writing

YouTube / Video Resources:

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