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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE

R26 Regulation

B. TECH I & II Year COURSE STRUCTURE AND SYLLABUS

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

INSTITUTE VISION

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

INSTITUTE MISSION

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

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

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

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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT VISION

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

DEPARTMENT MISSION

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

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

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

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

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

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

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

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: **Analysis & Design:**

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

PSO2: **Computational Logic:**

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

PSO3: **Software Development:**

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

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

INDUCTION PROGRAMME

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

I B. Tech. – I Semester (CSE)

S.No.	Course Code	Subject	L	T	Pr	P	Credits
1	26HS0809	Applied Chemistry for Computer Science	3	0	0	0	3
2	26HS0866	Linear Algebra And Calculus	3	0	2	0	4
3	26ME0303	Engineering Graphics	1	0	4	0	3
4	26CS0501	Computational Thinking and Problem Solving with Python	3	0	0	0	3
5	26CS3101	AI Tools and Applications	1	0	4	0	3
6	26CS0502	Computational Thinking and Problem Solving with Python Lab	0	0	0	3	1.5
7	26HS0810	Applied Chemistry for Computer Science Lab	0	0	0	3	1.5
8	26ME0302	Engineering Workshop	0	0	0	3	1.5
9	26HS0823	NSS/NCC/Scouts & Guides/Community Service	0	0	0	1	0.5
Total			**11**	**0**	**10**	**10**	**21**

I B. Tech. – II Semester (CSE)

S.No.	Course Code	Subject	L	T	Pr	P	Credits
1	26HS0821	Communicative English	2	0	0	0	2
2	26HS0849	Physics for Advanced Computing	3	0	0	0	3
3	26HS0867	Differential Equations and Vector Calculus	3	0	2	0	4
Professional core-I							
4	26CS0503	Data Structures & Efficient Problem Solving	2	0	2	0	3
5	26ME0301	Design Thinking	1	0	4	0	3
6	26CS0504	Data Structures & Efficient Problem Solving Lab	0	0	0	2	1
7	26HS0850	Physics for Advanced Computing Lab	0	0	0	3	1.5
8	26HS0822	Communicative English Lab	0	0	0	2	1
9	26HS0824	Health and wellness, Yoga and Sports	0	0	0	1	0.5
10	26CS4801	Cybersecurity Awareness for Engineers	2	0	0	0	0
Total			**14**	**0**	**8**	**8**	**19**

II B. Tech. – I Semester (CSE)

S.No.	Course Code	Subject	L	T	Pr	P	Credits
1	26HS0869	Probability and Statistics	2	0	2	0	3
2	26HS0825	Universal Human Values – Understanding Harmony and Ethical Human Conduct	3	0	0	0	3
3	26ME0305	Managerial Economics and Financial Analysis	2	0	0	0	2
4	26CS0505	Digital Logic and Computer Organization	3	0	0	0	3
Professional Core - II							
5	26CS0506	Algorithms Design and Analysis	3	0	0	0	3
Professional Core - III							
6	26CS0507	Object Oriented Design and Programming	3	0	0	0	3
7	26CS0508	Digital Logic and Computer Organization Lab	0	0	0	2	1
Professional Core - II Lab							
8	26CS0509	Algorithms Design and Analysis Lab	0	0	0	3	1.5
Professional Core - III Lab							
9	26CS0510	Object Oriented Design and Programming Lab	0	0	0	3	1.5
Skill Enhancement Course I							
10	26CS0570	AI Assisted Application Development	1	0	0	2	2
11	26HS0812	Environmental Science	2	0	0	0	0
Total			**20**	**0**	**2**	**10**	**23**

II B. Tech. – II Semester (CSE)

S.No.	Course Code	Subject	L/D	T	Pr	P	Credits
1	26HS0811	Biology for Engineers	2	0	0	0	2
2	26CE0107	Sustainability and Green Engineering	2	0	0	0	2
Professional Core - IV							
3	26CS0511	Operating Systems and Intelligent Resource Management	3	0	0	0	3
Professional Core - V							
4	26CS0512	Intelligent Database Management Systems	3	0	0	0	3
Professional Core - VI							
5	26CS0513	Automata Theory And Compiler Design	2	0	2	0	3
Professional Core - VII							
6	26CS0514	Software Engineering for AI Driven System	2	0	2	0	3
Professional Core - IV Lab							
7	26CS0515	Operating Systems and Intelligent Resource Management Lab	0	0	0	3	1.5
Professional Core - V Lab							
8	26CS0516	Intelligent Database Management Systems Lab	0	0	0	3	1.5
Skill Enhancement Course II							
9	26CS0571	Data Analytics Using Python	1	0	0	2	2
10	26HS0826	Technical Paper Writing & IP	2	0	0	0	0
Total			**15**	**0**	**4**	**8**	**21**

YEAR	I		II		III		IV		TOTAL
SEM	I	II	I	II	I	II	I	II	
CREDITS	21	19	23	21					84

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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
List of Subjects

S.No.	Subject Code	Name of Subject
Core Subjects		
1	26HS0809	Applied Chemistry for Computer Science
2	26HS0866	Linear Algebra And Calculus
3	26ME0303	Engineering Graphics
4	26CS0501	Computational Thinking and Problem Solving with Python
5	26CS3101	AI Tools and Applications
6	26CS0502	Computational Thinking and Problem Solving with Python Lab
7	26HS0810	Applied Chemistry for Computer Science Lab
8	26ME0302	Engineering Workshop
9	26HS0823	NSS/NCC/Scouts & Guides/Community Service
10	26HS0821	Communicative English
11	26HS0849	Physics for Advanced Computing
12	26HS0867	Differential Equations and Vector Calculus
Professional core-I		
13	26CS0503	Data Structures & Efficient Problem Solving
14	26ME0301	Design Thinking
15	26CS0504	Data Structures & Efficient Problem Solving Lab
16	26HS0850	Physics for Advanced Computing Lab
17	26HS0822	Communicative English Lab
18	26HS0824	Health and wellness, Yoga and Sports
19	26CS4801	Cybersecurity Awareness for Engineers
20	26HS0869	Probability and Statistics
21	26HS0825	Universal Human Values – Understanding Harmony and Ethical Human Conduct
22	26ME0305	Managerial Economics and Financial Analysis
23	26CS0505	Digital Logic and Computer Organization
Professional Core - II		
24	26CS0506	Algorithms Design and Analysis
Professional Core - III		

25	26CS0507	Object Oriented Design and Programming
26	26CS0508	Digital Logic and Computer Organization Lab
Professional Core - II Lab		
27	26CS0509	Algorithms Design and Analysis Lab
Professional Core - III Lab		
28	26CS0510	Object Oriented Design and Programming Lab
Skill Enhancement Course I		
29	26CS0570	AI Assisted Application Development
30	26HS0812	Environmental Science
31	26HS0811	Biology for Engineers
32	26CE0107	Sustainability and Green Engineering
Professional Core - IV		
33	26CS0511	Operating Systems and Intelligent Resource Management
Professional Core - V		
34	26CS0512	Intelligent Database Management Systems
Professional Core - VI		
35	26CS0513	Automata Theory And Compiler Design
Professional Core - VII		
36	26CS0514	Software Engineering for AI Driven System
Professional Core - IV Lab		
37	26CS0515	Operating Systems and Intelligent Resource Management Lab
Professional Core - V Lab		
38	26CS0516	Intelligent Database Management Systems Lab
Skill Enhancement course – II		
39	26CS0571	Data Analytics Using Python
40	26HS0826	Technical Paper Writing & IPR

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY: PUTTUR
(AUTONOMOUS)

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	0	0	3

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

Prerequisites: Basic Chemistry (Higher Secondary level)

Course Objectives :

The objectives of the course is to

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

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

Unit – II :	**Semiconductor Chemistry**
Content	Atomic structure and chemical bonding relevant to semiconductors. Band theory: valence band, conduction band, band gap energy. Intrinsic and extrinsic semiconductors: silicon, germanium, gallium arsenide. Doping

	chemistry: n-type (phosphorus, arsenic) and p-type (boron, aluminium) dopants. Chemical vapour deposition (CVD) for thin film formation; silicon dioxide (SiO_2) formation and properties. Chemistry of compound semiconductors: GaAs, InP, SiC.
AI Integration	AI-based prediction of semiconductor properties using ML models; process parameter optimization in semiconductor fabrication using machine learning; Python-based band gap prediction using regression models. Free/Open-Source AI Tools: Python, TensorFlow, Scilab.

Unit–III :	**Electrochemistry and Energy Storage**
Content	Electrochemical cells: galvanic cells, electrode potential, Nernst equation and problems; potentiometry (redox reactions); conductometry (strong acid vs. strong base, weak acid vs. strong base). Primary batteries: Zinc-air battery, Sodium-air battery. Secondary batteries: chemistry of charging and discharging of lead-acid, nickel-cadmium, nickel-metal hydride. Lithium-ion batteries: construction, working, and applications of $LiCoO_2$. Supercapacitors: definition, classification, and applications. Fuel cells: construction, working, and applications of hydrogen fuel cells and direct methanol fuel cells (DMFC). Battery management systems: overcharge protection and thermal management.
AI Integration	AI models for battery health prediction and state-of-charge optimization using scikit-learn; data analysis for energy efficiency in data centres using Python (Pandas); ML-based prediction of electrochemical impedance spectra. Free/Open-Source AI Tools: Python (Pandas, scikit-learn), Google Colab.

Unit–IV :	**Chemistry of Corrosion and Protective Techniques**
Content	Corrosion: definition, examples, types chemical corrosion and electrochemical corrosion; Pilling-Bedworth rule. Galvanic corrosion; concentration cell corrosion; factors affecting corrosion. Deal-Grove model: thermal oxidation of silicon; passivation layers silicon dioxide, silicon nitride, and aluminium oxide. Corrosion of printed circuit boards (PCBs): copper oxidation, tin whiskers; corrosion in solder joints and interconnects. Protective techniques: proper design, using pure metals, metal alloys, inhibitors; cathodic protection sacrificial anodic protection, impressed current cathodic protection. Electroplating (nickel) and electroless plating (copper).
AI Integration	Computer vision (OpenCV) for automated defect detection in PCBs; AI-based prediction of corrosion failure in electronic components using TensorFlow; image classification models for identifying corrosion types from

	microscope images. Free/Open-Source AI Tools: OpenCV, TensorFlow, Python.

Unit –V :	Chemistry of Nanomaterials
Content	Introduction to nanochemistry: size-dependent properties, quantum confinement. Carbon nanomaterials: fullerenes, carbon nanotubes (CNTs), graphene. Synthesis methods: chemical vapour deposition (CVD), sol-gel process. Quantum dots: synthesis, properties, applications in displays and quantum computing. Metallic nanoparticles: preparation and applications of gold, silver, and copper nanoparticles. Applications: nanoelectronics, interconnects, sensors (optical and biosensors), memory devices. Spintronics materials: magnetic nanoparticles, topological insulators.Chemistry of quantum computing materials: superconducting materials (NbTi, Nb_3Sn); ion trap materials.
AI Integration	AI-based nanomaterial property prediction using machine learning (Python, TensorFlow); simulation of nano-devices and quantum materials using open-source tools; generative AI for exploring nanomaterial design spaces. Free/Open-Source AI Tools: Python, TensorFlow, open-source materials simulation platforms.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply quantum chemistry principles to analyze molecular systems and explain molecular orbital theory for computing materials.	**L3–L4**
CO2	Analyze semiconductor materials, doping chemistry, and fabrication processes used in modern electronic devices.	**L3–L4**
CO3	Evaluate energy storage technologies including batteries, supercapacitors, and fuel cells for computing and data centre applications.	**L3–L4**
CO4	Analyze corrosion mechanisms in electronic devices and recommend appropriate protective techniques for PCBs and interconnects.	**L3–L4**
CO5	Design advanced materials solutions using nanotechnology principles and AI-assisted tools for nanoelectronics and quantum computing.	**L5–L6**

Bloom's Taxonomy Coverage – Summary

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

CO – PO / PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

1. Müller, A., & Cheetham, A. K. (Eds.). *The Chemistry of Nanomaterials: Synthesis, Properties and Applications* (Vols. 1–2). Wiley-VCH, 2007.
2. Poole, C. P., & Owens, F. J. *Introduction to Nanotechnology*. John Wiley & Sons, 2003.
3. Sharma, B. K., & Gaur, H. *Engineering Chemistry*. Krishna Prakashan Media, 2019.
4. Dara, S. S., & Umare, S. S. *Engineering Chemistry* (15th Ed.). S. Chand Publishing, 2018.

Online Resources :

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	2	0	4

(26HS0866) LINEAR ALGEBRA AND CALCULUS

(Common to all branches)

Prerequisites: Higher Secondary Mathematics

Course Objectives :

The objectives of the course is to

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

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

Unit – II :	**Eigenvalues, Eigenvectors, Diagonalization and Quadratic Forms**
Content	Eigenvalues and eigenvectors; characteristic equation; Cayley-Hamilton theorem; diagonalization; similarity transformation; powers and inverse of

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

Unit – III :	**Single Variable Calculus and Series Expansions**
Content	Rolle's theorem; Lagrange's mean value theorem; Cauchy's mean value theorem; Taylor's and Maclaurin's series with remainder; approximation of functions; error estimation. Curvature of plane curves; radius and centre of curvature; circle of curvature; applications to engineering approximations and curve design.
AI Integration	SymPy-based Taylor and Maclaurin expansion generation; plotting truncation error and convergence; AI-supported derivation checking for mean value theorems; curvature plot visualization and interpretation. Free/Open-Source AI Tools: Python (SymPy, NumPy, Matplotlib); Google Colab; Jupyter Notebook; WolframAlpha free tier.

Unit – IV :	**Multivariable Differentiation and Optimization**
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**	
Content	Double integrals; evaluation over type I and type II regions; change of order of integration; triple integrals; change of variables; polar, cylindrical and	

	spherical coordinates. Area and volume computation; mass, centroid and moment calculations; engineering applications of multiple integration.
AI Integration	Numerical evaluation of double and triple integrals using SciPy; 2D and 3D region visualization; AI-assisted generation of integration code; interpreting Jacobian as geometric scaling factor in transformed coordinates. Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib); Google Colab; SymPy; Jupyter Notebook; 3D plotting tools.

Course Outcomes (COs)

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

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

Bloom's Taxonomy Coverage – Summary

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

CO – PO Articulation Matrix (NBA)

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

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

CO4	3	3	3	3	2	-	-	-	1	2	-	-
CO5	3	3	2	3	3	-	1	1	2	2	1	-
CO-PO Correlation Highlights	CO1 and CO2 strongly align with PO1 (Engineering Knowledge) and PO2 (Problem Analysis) through matrix and eigenvalue problem-solving. CO4 and CO5 map to PO3 and PO4 through optimization and multiple integral applications. PO5 (Modern Tools) is addressed throughout via Python, SymPy, and Google Colab integration. CO5 additionally links to PO7 and PO8 through engineering field interpretation, and strongly to PO12 (Lifelong Learning) through continued computational skill development beyond the course.											

Textbooks :

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

Reference Books :

1. G. B. Thomas, M. D. Weir and J. Hass, *Thomas' Calculus*, 14th Edition, Pearson, 2018.
2. R. K. Jain and S. R. K. Iyengar, *Advanced Engineering Mathematics*, 5th Edition, Alpha Science International, 2021.
3. Gilbert Strang, *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2016.
4. D. C. Lay, S. R. Lay and J. J. McDonald, *Linear Algebra and Its Applications*, 5th Edition, Pearson, 2015.
5. G. B. Arfken, H. J. Weber and F. E. Harris, *Mathematical Methods for Physicists*, 7th Edition, Academic Press, 2012.
6. D. G. Zill and W. S. Wright, *Advanced Engineering Mathematics*, 6th Edition, Jones & Bartlett Learning, 2018.

Online Resources :

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

Recommended Free & Open-Source AI/Computational Tools :

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

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
1	0	4	0	3

(26ME0303) ENGINEERING GRAPHICS
(Common to all branches)

Prerequisites: Basic Geometry, Fundamental Python Programming

Course Objectives :

The objectives of this course is to

1. *Enable the students with various concepts like dimensioning, conventions and standards relate to Engineering Drawing*
2. *Impart knowledge on the projection of points, lines and plane surfaces.*
3. *Improve the visualization skills for better understanding of projection of solids.*
4. *Develop the imaginative skills of the students required to understand Section of solids and Developments of surfaces.*
5. *Make the students understand the viewing perception of a solid object in Isometric and Perspective projections.*

List of Experiments :

Unit-I :	Introduction to Engineering Graphics, Curves and Scales

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

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

3. **Construction of Plain, Diagonal and Vernier Scales**
   Construct a plain scale, a diagonal scale, and a vernier scale for given representative fractions and measure specified distances using each. Cross-verify readings between manually constructed scales and a digital scale tool.

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

Unit-II :	Orthographic Projections of Points, Lines and Planes

1. **Projection of Points in All Four Quadrants**
   Draw the orthographic projections (top view and front view) of points located in each of the four quadrants relative to the horizontal and vertical reference planes. Verify the position of each projection against the given point coordinates.
2. **Projection of Lines Inclined to Reference Planes**
   Determine the projections of a straight line inclined to one or both reference planes; find the true length and true inclination of the line using the rotation or auxiliary plane method. Cross-check results using trigonometric calculation.
3. **Projection of Planes in Various Orientations**
   Draw projections of regular planes (triangle, square, pentagon, hexagon) positioned perpendicular to both reference planes, parallel to one reference plane, and inclined to both reference planes. Verify true shape recovery using the auxiliary view method.
4. **3D Coordinate Estimation Script from 2D Projections**
   Develop a transformation script where a user inputs 2D projection endpoints (top view and front view coordinates) and the script automatically computes the 3D line representation, true spatial length, and inclination angles. Validate the script output against a manually solved projection problem.

Unit-III :	Projection of Regular Solids

1. **Projection of Solids with Axis Perpendicular to a Reference Plane**
   Draw the projections of a hexagonal prism and a cone with the axis perpendicular to the horizontal plane, and repeat with the axis perpendicular to the vertical plane. Verify true shape and hidden line conventions in each case.
2. **Projection of Solids with Axis Inclined to a Reference Plane**
   Draw the projections of a square pyramid with its axis inclined to one reference plane and parallel to the other, using the change-of-position or auxiliary plane method. Verify the apparent and true lengths of the solid's edges.
3. **Solid Classification Using a Machine Learning Notebook**
   Set up a classification notebook using scikit-learn or a simple convolutional model to automatically identify the type of solid (e.g., hexagonal prism vs. cone) present within a given orthographic engineering layout image. Test the classifier against five different solid images and document accuracy.

Unit-IV :	Sections of Solids and Development of Surfaces

1. **Sectional Views of Solids with Inclined Cutting Planes**
   Draw the sectional view and true shape of section for a pyramid and a cone cut by a section plane inclined to the base, for solids in simple position. Verify the true shape of the cut surface using the auxiliary projection method.
2. **Development of Lateral Surfaces Parallel Line Method**

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

3. **Development of Lateral Surfaces Radial Line Method**
   Develop the lateral surface of a cone and a pyramid using the radial-line development method. Verify the unfolded pattern by reconstructing a physical or digital model from the developed surface.
4. **Sheet Nesting Optimization Using Heuristic Algorithms**
   Use a nesting optimization script (genetic algorithm or standard heuristic) to arrange multiple custom flat-surface solid developments onto a finite raw sheet grid, minimizing material margins and cut lines. Compare the optimized layout with a manually arranged layout in terms of material utilization.

Unit-V :	Isometric Projections and Computer Graphics

1. **Isometric Projection of Simple and Compound Solids**
   Construct the isometric view of a simple solid (cube, prism, pyramid) using the isometric scale, and extend to a compound solid formed by combining two basic solids. Verify isometric axis angles and dimensional accuracy.
2. **Conversion Between Orthographic and Isometric Views**
   Given a set of orthographic views (front, top, side) of an object, construct the corresponding isometric view; conversely, given an isometric view, derive the orthographic views. Cross-verify both conversions for dimensional consistency.
3. **Computer-Aided Drafting Using AutoCAD**
   Create 2D drawings of simple machine components and a basic PCB layout using AutoCAD; apply transformation commands (move, rotate, scale, mirror) to manipulate the drawn entities. Export and document the final drawings (not for end examination).
4. **2D-to-3D Depth Reconstruction Using a Pretrained Model**
   Use a pretrained depth-estimation or pixel-to-voxel neural model to parse a set of front, top, and side drawings, compiling them into an interactive 3D point cloud or boundary surface rendering. Compare the AI-reconstructed 3D model with the manually constructed isometric drawing.

AI Tools Integration (Lab)

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

Lab Course Outcomes

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

CO	Course Outcome Statement	Bloom's Level
CO1	Construct engineering curves and scales using traditional drafting geometry validated by automated algorithmic paths.	**L3 (Apply)**
CO2	Solve multi-planar projection instances of linear points, segments, and standard regular planar silhouettes.	**L4 (Analyze)**
CO3	Render projections of standard 3D solids and identify variations using structural boundary algorithms.	**L4 (Analyze)**
CO4	Draft complex cross-sectional geometries and mapped surface extensions of sliced geometric forms.	**L4 (Analyze)**
CO5	Produce composite isometric layouts alongside generative text-to-CAD system alternatives for dynamic prototyping.	**L5 (Evaluate)**

CO – PO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources

- NPTEL Engineering Graphical Design Course – IIT Kharagpur
- OpenCV Official Python Bootcamps and Tutorials
- MIT 6.S192: Deep Generative Modeling Platforms

Video Links & Online Lectures

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0501) COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON

(Common to all branches)

Course Objectives :

The objectives of the course is to

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

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

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

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

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

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

Unit – V :	**File Handling, Exception Handling, And Oop**
Content	Modules and packages: creating and importing modules; standard library modules. File handling: opening, reading, writing, and closing files; file modes; working with text files; processing CSV and Excel files.

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

Course Outcomes (Cos):

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

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

Bloom's Taxonomy Coverage – Summary

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

CO – PO / PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

1. Algorithm & Flowchart: Flowgorithm , flowgorithm.org,
2. Lucidchart , lucidchart.com ,
3. Algorithm Visualizer , algorithm-visualizer.org
4. Programming & Debugging: Python Tutor – pythontutor.com
5. Thonny – thonny.org
6. Replit – replit.com
7. Data & Scientific: Pandas – pandas.pydata.org
8. NumPy – numpy.org
9. SymPy – sympy.org
10. NLP & Text: NLTK – nltk.org
11. TextBlob – textblob.readthedocs.io
12. Development: Jupyter Notebook – jupyter.org
13. PyCharm – jetbrains.com/pycharm
14. OpenPyXL – openpyxl.readthedocs.io
15. NPTEL – Problem Solving through Programming in C /
16. Python – nptel.ac.in

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

I B.Tech – I Sem.

L	T	Pr	P	C
1	0	4	0	3

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

Course Objectives :

The objectives of the course is to

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

Unit – I :	**Understanding AI And AI For Documents And Presentations**
Content	What is AI: AI vs. ML vs. Deep Learning vs. Generative AI; brief history (Turing Test → ChatGPT → generative AI era); how LLMs work (training data, next-token prediction, temperature, hallucinations, context windows, knowledge cutoffs). AI ecosystem: ChatGPT, Claude, Gemini, Microsoft Copilot strengths and differences. AI for presentations: Gamma.app full slide deck from a paragraph prompt; Microsoft Copilot in PowerPoint slide generation, design suggestions, speaker notes; Canva AI Magic Design, text-to-presentation, brand kits; Beautiful.ai smart slide layouts; comparing AI-generated vs. manually created presentations. AI for document writing: Microsoft Copilot in Word draft, rewrite, summarize; Gemini in Google Docs writing assistance and inline Q&A; Notion AI meeting notes, project plans, knowledge base; Claude Projects organizing documents and having a persistent AI assistant over a set of files. AI writing assistance: Grammarly grammar, tone, clarity, plagiarism; ChatGPT and Claude for lab reports, proposals, and letters; academic integrity AI as an assistant not an author; disclosure norms; identifying AI-generated content using GPTZero.
AI Integration	Use Gamma.app to generate a presentation on an engineering topic in under 2 minutes; use Copilot in Word or Gemini in Docs to draft and improve a report;

	compare ChatGPT, Claude, Gemini, and Microsoft Copilot on the same writing task; use Grammarly to review your own writing.

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

Unit – III :	AI For Visual Design, Diagrams, And Media
Content	AI for image generation: DALL-E 3 (via ChatGPT), Canva AI (text-to-image, Magic Design, background remover), Adobe Firefly, Ideogram writing effective image prompts (style, subject, composition, colour); iterative prompt refinement for posters, diagrams, and mockups. AI for diagrams and visual thinking: Napkin.ai converting text or concepts into visual diagrams, charts, and infographics automatically; Whimsical AI AI-generated mind maps, flowcharts, and wireframes from a description; Miro AI collaborative whiteboard with AI summarization and diagramming. AI for audio and video: ElevenLabs text-to-speech and voice cloning for presentations and voiceovers; Suno and Udio AI music generation for project videos; Runway and Pika generating short video clips from text prompts; HeyGen AI avatars for video presentations; current limitations and appropriate use cases.

	AI content detection and critical evaluation: GPTZero, Originality.ai for AI-text detection; Google Lens and reverse image search for AI-generated images; fact-checking workflows cross-referencing primary sources; deepfakes and synthetic media recognition, risks, and responsible creation.
AI Integration	Use Napkin.ai to convert a paragraph of text into a diagram; use Whimsical AI to generate a mind map of a course topic; use DALL-E 3 or Canva AI to create an event poster through 3 prompt iterations; use ElevenLabs to create a voiceover; test GPTZero on 5 samples and analyse confidence scores.

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

Unit – V :	**Responsible AI, Ethics, And Capstone Project**
Content	AI bias and fairness: sources of bias (data, model, deployment); real documented cases Amazon hiring algorithm (2018), COMPAS recidivism (2016), facial recognition disparities (NIST 2019); fairness metrics; misinformation, deepfakes, and synthetic media risks. Privacy and intellectual property: what happens to data entered into public AI tools; PII risks; data retention policies of ChatGPT (OpenAI), Claude (Anthropic), and Gemini (Google); who owns AI-generated content;

	copyright status globally; academic plagiarism with AI; personal data safety protocol. AI governance frameworks: EU AI Act risk tiers (unacceptable, high, limited, minimal); UNESCO AI Ethics Recommendation; India's NITI Aayog Responsible AI Guidelines; India Digital Personal Data Protection Act 2023 (DPDP Act); environmental cost of AI energy and carbon footprint of LLM training and inference. Capstone Mini-Project: each student (or pair) defines a real engineering problem, selects at least 3 AI tool categories from UNITs I–IV, produces a quality-verified deliverable (report, presentation, diagram, or automation), and presents in 5 minutes. Evaluation: problem clarity, tool selection, output quality, human refinement, responsible AI disclosure.
AI Integration	Read and compare data retention policies of ChatGPT, Claude, and Gemini; classify 5 daily-use AI systems using EU AI Act risk tiers; use GPTZero and Google Lens to detect AI-generated content; design and present the Capstone Mini-Project integrating tools from all five UNITs.

Course Outcomes (Cos) :

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

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

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities / Assessments

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

CO – PO / PSO Articulation Matrix (NBA)

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

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

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

Reference Books:

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

Online Resources :

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0502) COMPUTATIONAL THINKING AND PROBLEM SOLVING WITH PYTHON LAB

(Common to all branches)

Prerequisites: None – Foundation Course. (Inferred – not specified in the source.)

Course Objectives :

The objectives of the course is to

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

List of Experiments

MODULE I - Computational Thinking & Programming Basics :

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

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

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

Experiments:

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

MODULE – II Decision Making & Looping:

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

AI Integration: Generate boundary and test cases; verify decision logic; trace loop execution; compare loop alternatives; optimize nested loops.
Suggested Tools: Python, Python Tutor, ChatGPT, Microsoft Copilot
Experiments:

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

MODULE III - Strings & Data Structures :

Topics: String processing; lists and nested lists; tuples and sets.
AI Integration: Generate string test cases and sample datasets; explain slicing and immutable structures; compare list approaches and suggest optimizations.
Suggested Tools: Python, Python Tutor, Algorithm Visualizer, ChatGPT
Experiments:

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

MODULE IV - Functions & Problem Solving :

Topics: User-defined functions; recursion; dictionaries and lambda functions.
AI Integration: Generate function examples and datasets; verify outputs; generate recursion trees; compare iterative and recursive solutions; explain lambda execution.
Suggested Tools: Python, Python Tutor, Algorithm Visualizer, ChatGPT
Experiments:

1. **Experiment 10: User-Defined Functions** Implement positional, keyword, default, and variable-length arguments.
2. **Experiment 11: Recursive Problem Solving** Develop recursive solutions for factorial, Fibonacci, GCD, and Tower of Hanoi.
3. **Experiment 12: Dictionary and Lambda Applications** Implement a student database, dictionary operations, dictionary sorting, and lambda-based computations.

MODULE V - File Handling, Exception Handling & Oop:

Topics: File processing and data storage; exception handling and testing; object-oriented programming classes, objects, constructors, methods.
AI Integration: Generate datasets and error scenarios; verify file outputs; recommend exception strategies; generate class diagrams and suggest object interactions.

Suggested Tools: Python, ChatGPT, Microsoft Copilot

Experiments:

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

Suggested Mini Projects:

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

Course Outcomes (COs)

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

CO – PO Articulation Matrix

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

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

Textbooks :

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

Reference Books :

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

Recommended AI / Computational Tools :

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

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

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

Course Objectives :

The objectives of the course is to

1. *To develop practical skills in performing basic chemistry laboratory experiments safely and accurately.*
2. *To understand the properties and applications of engineering materials, including polymers, nanomaterials, and advanced materials relevant to computer science.*
3. *To analyze water quality and learn techniques for water treatment used in industrial and technological applications.*
4. *To gain hands-on experience with instrumental and analytical techniques, and interpret experimental results for engineering applications.*

List of Experiments :

Unit-I :	Titrimetric and Conductometric Analysis

1. **Estimation of Ferrous Iron (Fe2+) using Potassium Permanganate (KMnO4) Redox Titration**
   Standardize a KMnO4 solution against a known oxalate solution, then titrate a Fe2+ sample solution against the standardized KMnO4 until a persistent pink endpoint is observed. Calculate the concentration and amount of Fe2+ present in the given sample using stoichiometric relations.
2. **Determination of Strength of Acid (H2SO4) in Battery Using Sodium Hydroxide (NaOH)**
   Standardize a NaOH solution against a primary standard, then titrate a diluted battery acid sample against the standardized NaOH using a suitable indicator. Determine the strength and concentration of sulfuric acid present in the battery sample.
3. **Determination of Corrosion**
   Expose metal coupons (mild steel or copper) to a corrosive medium for a fixed duration, measure weight loss before and after exposure, and calculate the corrosion rate using the weight-loss method. Document observations on surface changes.
4. **Determination of Corrosion by Environment**
   Expose identical metal coupons to different environmental conditions (humid, saline, acidic) and compare corrosion rates across environments. Analyse which

environmental factor accelerates corrosion most significantly and relate findings to real-world electronic hardware protection needs.

Unit-II :	Conductometric and Potentiometric Titrations

1. **Conductometric Titration of Strong Acid vs Strong Base (HCl vs NaOH)**
   Titrate HCl against NaOH while measuring conductivity at regular intervals using a conductivity meter. Plot conductance versus volume of titrant added, identify the equivalence point from the graph, and determine the concentration of the acid.
2. **Conductometric Titration of Weak Acid vs Strong Base (CH3COOH vs NaOH)**
   Titrate acetic acid against NaOH while recording conductivity readings. Plot the conductometric titration curve, identify the characteristic curve shape for a weak acid–strong base titration, and determine the equivalence point and acid concentration.
3. **Potentiometry (Redox Titration) Estimation of Iron (Fe2+) Using Potassium Dichromate (K2Cr2O7)**
   Set up a potentiometric titration using a platinum electrode and reference electrode, titrate a Fe2+ solution against standardized K2Cr2O7, and record EMF values at regular intervals. Plot EMF versus volume to identify the equivalence point and calculate Fe2+ concentration.

Unit-III :	Spectroscopic Analysis and Polymer Synthesis

1. **Spectrophotometry Determination of Wavelength (λ) and Verification of Beer-Lambert's Law**
   Prepare a series of standard solutions of a coloured compound at varying concentrations, measure absorbance at a fixed wavelength using a UV-Vis spectrophotometer, and plot absorbance versus concentration to verify Beer-Lambert's law. Determine the molar extinction coefficient from the slope.
2. **Identification of Functional Groups in Organic Compound by Infra-Red (IR) Spectrum**
   Record the IR spectrum of a given organic compound and identify characteristic absorption bands corresponding to functional groups (C=O, O-H, N-H, C-H, etc.). Compare the spectrum with reference IR charts to confirm the identity of functional groups present.
3. **Preparation of Polymer (Phenol-Formaldehyde Bakelite)**
   Synthesize Bakelite by the controlled condensation reaction of phenol and formaldehyde under acidic or basic catalytic conditions, observing the resin formation stages (A-stage, B-stage). Document the reaction conditions and resulting polymer properties, and analyse the reaction's industrial relevance to electronic component insulation.

AI Tools Integration (Lab) :

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

Lab Course Outcomes

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

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

CO – PO Articulation Matrix :

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

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	2	-	1	2	-	-	-	-	-	3	2
CO2	2	-	-	2	-	2	1	-	-	-	-	-	2	3
CO3	2	-	-	3	-	3	2	-	-	-	-	-	2	3
CO4	2	-	-	2	-	2	1	-	-	-	-	-	3	2
CO5	1	-	-	2	3	2	2	-	-	-	-	-	2	3
Average	2.0	-	-	2.2	3.0	2.0	1.6	-	-	-	-	-	2.4	2.6

CO-PO Correlation Highlights	CO1 and CO3 map strongly to PO1 and PO2 through titration-based quantitative analysis. CO2 connects to PO4 (Investigation) through corrosion rate determination under varying environmental conditions.

	CO5 links to PO5 (Modern Tools) through Python-based predictive modelling of material behaviour and polymer synthesis documentation.

Textbooks:

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

Reference Books :

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

Online Resources :

1. IEEE Xplore Digital Library
2. Semiconductor Industry Association resources
3. NPTEL Engineering Chemistry

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 :

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

Lab Software & AI Tools Required:

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

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

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

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

Prerequisites: Basic Computing Elements

Course Objectives :

The objectives of the course is to

1. *To familiarize students with basic workshop practices and the safe handling of tools, equipment, and machines.*
2. *To develop practical skills in fitting, carpentry, welding, sheet metal, and electrical workshop operations.*
3. *To enhance the ability to read, interpret, and fabricate simple engineering components using standard workshop techniques.*
4. *To build confidence in manufacturing, assembly, and maintenance practices, fostering problem-solving and teamwork skills in engineering environments.*

List of Experiments :

Unit-I :	Safety Practices, Carpentry and Fitting Trades

1. **Workshop Safety Orientation and PPE Demonstration**
   Demonstrate proper use of Personal Protective Equipment (helmet, safety glasses, gloves, apron) and identify common workshop hazards. Practise emergency protocols including fire extinguisher use and first-aid station location, using computer vision-based safety compliance monitoring as a demonstration.
2. **Carpentry Half-Lap and Mortise & Tenon Joints**
   Prepare a half-lap joint and a mortise-and-tenon joint using basic carpentry tools (saw, chisel, mallet, marking gauge) on a wooden workpiece. Verify joint fit and dimensional accuracy against the given drawing specification.
3. **Carpentry Corner Dovetail or Bridle Joint**
   Prepare a corner dovetail joint or bridle joint following standard carpentry marking and cutting procedures. Inspect the finished joint for fit, squareness, and surface finish.
4. **Fitting V-Fit and Dovetail Fit**
   Perform a V-fit and a dovetail fit exercise on mild steel workpieces using fitting tools (files, scriber, try square, vernier caliper). Verify dimensional accuracy and fit tolerance against the drawing.
5. **Fitting Semi-Circular Fit**

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

Unit-II :	Sheet Metal, Plumbing and Welding Trades

1. **Sheet Metal Working Tapered Tray and Conical Funnel**
   Develop and fabricate a tapered tray and a conical funnel from GI sheet using sheet metal tools (snips, mallet, stakes); apply development of surfaces principles for accurate pattern marking. Verify dimensional accuracy of the finished components.
2. **Sheet Metal Working Elbow Pipe and Brazing**
   Develop and fabricate an elbow pipe joint from GI sheet, and perform a brazing exercise to join two metal pieces. Inspect the joint strength and surface finish of the brazed connection.
3. **Electrical Wiring Series, Parallel and Two-Way Switch Circuits**
   Make electrical connections for a parallel and series lighting circuit, and wire a two-way switch circuit for staircase lighting control. Test each circuit for correct operation and safety compliance using a multimeter.
4. **Electrical Wiring Godown Lighting, Tube Light and Soldering**
   Wire a godown lighting circuit and a tube light circuit (including choke and starter connections); practise soldering of wire joints. Test all connections for continuity and correct functioning.
5. **Plumbing Pipe Joint Preparation**
   Prepare a simple PVC pipe joint with coupling for same-diameter pipes and with a reducer for different-diameter pipes, using thread cutting and pipe-fitting tools. Test the joint for leak-tightness using a pressure or water test.
6. **Welding Shop Lap Joint and Butt Joint**
   Demonstrate and practise Arc Welding and Gas Welding techniques to prepare a simple lap joint and a butt joint on mild steel plates. Inspect the welded joint for uniformity, penetration, and visual defects.

Unit-III :	Foundry, Computer Hardware and Smart Systems

1. **Foundry Trade Green Sand Mould Preparation**
   Demonstrate and practise moulding tools and processes to prepare a green sand mould for a simple pattern. Inspect the mould cavity for surface finish and dimensional accuracy before pouring simulation.
2. **Introduction to 3D Printing**
   Demonstrate the core concepts of 3D printing including hardware components, software workflow, digital file preparation (STL/G-code), and printing materials. Observe the 3D printing process for a simple complex-geometry object and document the print quality.
3. **Computer Hardware Disassembly and Reassembly**
   Disassemble and reassemble a desktop computer to understand the role of each component (CPU, motherboard, RAM, HDD/SSD, GPU). Document the function and

interconnection of each component and verify successful reassembly through a boot test.

4. **Hardware Troubleshooting and Peripheral Installation**
   Simulate common hardware issues (RAM failure, CPU overheating) and identify the procedures for resolving each. Install and configure peripheral devices such as a printer, scanner, or external hard drive, and verify successful operation.
5. **Network Setup and Configuration**
   Set up a small LAN network with routers, switches, and computers; configure IP addresses, subnet masks, and basic network services. Verify network connectivity between devices using ping tests.
6. **Disassembly and Assembly of Mechanical Equipment**
   Familiarise with the disassembly and assembly process of a simple mechanical system such as a bicycle gear box or braking system. Document the function of each component and verify correct reassembly through operational testing.
7. **Smart Energy Monitoring Using IoT Sensors**
   Set up an IoT current sensor to monitor energy consumption of a simple electrical circuit; log the data and build a basic machine learning load forecasting model. Train an anomaly detection routine to flag unusual electrical load patterns (simulated arc/leakage events).

AI Tools Integration (Lab)

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

Lab Course Outcomes :

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

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

CO – PO Articulation Matrix (NBA)

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

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

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

Textbooks:

1. S. K. Hajra Choudhury, *Workshop Technology*, Volumes I & II, Media Promoters & Publishers.
2. B. S. Raghuwanshi, *A Course in Workshop Technology*, Dhanpat Rai & Co., 2017.

Reference Books :

1. Giacomo Veneri and Antonio Capasso, *Hands-On Industrial Internet of Things*, Packt Publishing, 2018.
2. Felix W., *Basic Workshop Technology: Manufacturing Process*, Independently Published, 2019.
3. Bruce J. Black, *Workshop Processes, Practices and Materials*, 5th Edition, Routledge, 2015.
4. P. N. Rao, *Manufacturing Technology: Foundry, Forming and Welding*, McGraw Hill Education.
5. R. Keith Mobley, *An Introduction to Predictive Maintenance*, 2nd Edition, Butterworth-Heinemann, 2002.

Online Resources:

1. NPTEL Workshop Practices Video Series – IIT Kharagpur
2. Predictive Maintenance Models in Python Tutorial
3. MIT 6.131: Power Electronics and Smart Grid Interfaces

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

I B.Tech – I Sem.

L	T	Pr	P	C
0	0	0	1	0.5

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

Course Objectives :

The objectives of the course is to

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

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

Unit-II :	**Leadership, Skill Development, Camping and Disaster Management**
Topics	Leadership development – theories and models (situational, transformational)

	Team building, communication & conflict resolution Parade, drill & physical training basics Map reading, navigation and field craft (NCC) Camp organisation – planning, logistics, safety protocols Adventure activities: trekking, rock-climbing, swimming (awareness) Disaster management – concepts, types, roles of NSS/NCC volunteers First Aid – CPR, wound management, fractures, snake bite, burn
Unit-wise Learning Outcomes (ULOs)	**On completion of this unit, students will be able to:** ULO2.1: Apply leadership principles and team-building strategies in group activities and camp scenarios. [Apply] ULO2.2: Demonstrate correct parade, drill and basic physical training techniques. [Apply] ULO2.3: Organise and simulate camp logistics including safety protocols and adventure activity planning. [Apply] ULO2.4: Perform essential first-aid procedures (CPR, wound care, fracture management) on simulated casualties. [Apply] ULO2.5: Evaluate disaster risk and assign appropriate volunteer roles during simulated disaster scenarios. [Evaluate]

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

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.

Course Outcomes (COs) :

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

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

CO – PO / PSO Mapping

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

CO / PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PS O2
CO 1	3	2	-	-	-	3	3	2	2	-	-	-	2	-

CO 2	-	-	3	-	-	3	2	3	3	-	-	-	2	2
CO 3	-	-	2	3	2	3	3	2	3	-	-	-	-	-
CO 4	-	-	3	2	-	2	2	3	3	-	-	-	-	-
CO 5	-	-	-	-	3	3	3	2	2	-	-	-	-	-

Text Books:

1. Ministry of Youth Affairs & Sports, Government of India, *NSS Training Manual*, MYAS Publications, 2022.
2. Directorate General, National Cadet Corps, *National Cadet Corps: Training Syllabus and Manual*, Directorate General NCC, New Delhi, 2021.

Reference Books:

1. Government of India, Ministry of Youth Affairs & Sports, *National Service Scheme Manual*, Government of India Publications, 2006.
2. Directorate General, National Cadet Corps, *NCC Handbook*, Directorate General NCC, New Delhi, 2020.
3. Satish Modh, *Disaster Management: A Holistic Approach*, 2nd Edition, Macmillan Publishers India, 2010.
4. Steve McCurley and Rick Lynch, *Volunteer Management: Mobilizing All the Resources of the Community*, Interpub Group, 2011.
5. Lord Robert Baden-Powell, *Scouting for Boys*, Oxford University Press, 2005.

Web Resources :

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

YouTube Resources :

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
2	0	0	0	2

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

Course Objectives :

The objectives of the course is to

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

UNIT-I : Foundations of English Communication

Topics Covered

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

Learning Outcomes

1. Recall the fundamental components and types of the communication process.

2. Identify and classify parts of speech and tense forms in given sentences.
3. Explain sentence structure patterns and apply them in original writing samples.
4. Articulate words with correct stress and intonation using digital phonetics tools.

AI Tools Integrated

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

Hands-On AI Activity

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

Unit-II : Listening and Speaking Skills

Topics Covered

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

Learning Outcomes

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

AI Tools Integrated

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

Hands-On AI Activity

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

Unit-III : Reading Comprehension & Writing Skills

Topics Covered

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

Unit Learning Outcomes :

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

AI Tools Integrated :

Tool 1	QuillBot	AI paraphrasing tool – rewrite passages at varying complexity levels
Tool 2	Hemingway Editor	Readability scoring, passive voice detection, and sentence simplification

Tool 3	ChatGPT	Essay outline generation, argument structuring, and content brainstorming
Tool 4	Rewordify	Vocabulary simplification for improving comprehension of complex texts

Hands-On AI Activity

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

Unit-IV : Professional & Business Communication

Topics Covered

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

Learning Outcomes

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

AI Tools Integrated

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

Hands-On AI Activity

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

Unit-V : Use of AI in Language Learning

Topics Covered

1. Artificial Intelligence in Communication, Meaning and basics of AI & Benefits and limitations of AI in human interaction
2. AI chatbots and virtual assistants in education: Use of AI in grammar correction and vocabulary building, AI-based speaking and pronunciation tools, writing assistants, E-Portfolio Creation, Students can also use general LLMs (like ChatGPT or Gemini) to summarize lengthy articles or generate texts on specific topics for reading comprehension.
3. AI in Global Communication: The use of automated translation, summarizing large texts, and finding information in different languages
4. (Suggested Tools: Tools like Google Translate and Reverso are used for real-time and document translation)
5. Digital Communication and AI Ethics: Responsible use of AI in education, Plagiarism and academic honesty, Fake information and AI-generated content, Privacy, bias, and ethical communication online

Unit wise Learning Outcomes :

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

AI Tools Integrated :

Tool 1	Copyleaks / Turnitin	Plagiarism detection, paraphrase checker, and academic integrity analysis
Tool 2	DeepL Translator	High-accuracy translation supporting intercultural communication understanding
Tool 3	LinkedIn Learning / Coursera	AI literacy modules – understanding AI bias, ethics, and responsible usage

Tool 4	Wix / Adobe Portfolio	E-portfolio creation platform for showcasing communication artifacts

Hands-On AI Activity :

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

Overall AI Tools Used in the Course :

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

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

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

Course Outcomes (COs) :

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

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

Code	Bloom's Level	Statement
CO6	Create	Students will be able to design and deliver professional communication artifacts , presentations, e-portfolios, and research reports , integrating AI tools for structure, style, and refinement.

Text Books :

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

Reference Books :

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

Online Learning Resources :

#	Resource Name	URL	Description
1	British Council – Learn English	https://learnenglish.britishcouncil.org	Free grammar, vocabulary, listening, and speaking practice modules
2	Coursera – Communication Skills	https://www.coursera.org/courses?query=communication	University-certified courses on professional and intercultural communication
3	edX – Writing and Communication	https://www.edx.org/learn/writing	Courses from MIT and Harvard on academic and business writing

#	Resource Name	URL	Description
4	Grammarly Blog	https://www.grammarly.com/blog	Grammar tips, writing guides, and professional communication resources
5	NPTEL – Communication Skills	https://nptel.ac.in/courses/109104101	IIT/IISc video lectures on English communication for engineering students
6	BBC Learning English	https://www.bbc.co.uk/learningenglish	Audio, video, and text resources for grammar, vocabulary, and pronunciation
7	Purdue OWL (Online Writing Lab)	https://owl.purdue.edu	APA/MLA citation guides, grammar rules, and professional writing resources

Recommended YouTube Resources :

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

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

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

CO	Bloom's Level	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	Remember	**1**	-	-	**1**	-	**1**	-	**1**	**1**	**3**	**1**	**2**
CO2	Understand	**1**	-	-	**1**	**1**	**2**	**1**	**1**	**2**	**3**	**1**	**2**
CO3	Apply	-	-	**1**	-	**3**	**2**	**1**	**2**	**3**	**3**	**2**	**2**
CO4	Analyze	-	**1**	**1**	**2**	**2**	**2**	**1**	**2**	**2**	**3**	**2**	**2**
CO5	Evaluate	-	**1**	**1**	**2**	**3**	**2**	**2**	**3**	**2**	**3**	**2**	**3**
CO6	Create	-	**1**	**2**	**2**	**3**	**3**	**2**	**3**	**3**	**3**	**3**	**3**
Avg		**1.0**	**1.0**	**1.3**	**1.6**	**2.4**	**2.0**	**1.4**	**2.0**	**2.2**	**3.0**	**1.8**	**2.3**

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

I B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

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

Prerequisites: Engineering Mathematics, Basic Electronics, Computer Science Fundamentals

Course Objectives :

The objectives of the course is

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

Unit– I :	**Wave Optics and Quantum Information Processing**
Content	Interference: Principle of superposition and coherence; Interference; Types of interference; Interference in thin films (reflection geometry) and Newton's rings. Applications: Determination of wavelength and refractive index. Diffraction: Introduction to diffraction; Fresnel and Fraunhofer diffraction; Fraunhofer diffraction due to single, double and N-slits (Diffraction Grating); Dispersive power and Resolving power of grating. Applications: CD/DVD data storage, Barcode scanners.Polarization: Types of polarization; Nicol's prism; Half-wave and Quarter-wave plates.
AI Integration	AI-assisted polarization management for quantum key distribution. AI-integrated holographic interferometry for surface flatness analysis. Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib and scikit-learn).

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

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

Unit– IV :	Semiconductors and Magnetic Materials
Content	Semiconductors: Introduction; Fermi energy and Fermi level in intrinsic and extrinsic semiconductors; Expression for concentration of electrons in conduction band and holes in valence band; Electrical conductivity; Expressions for drift and diffusion currents; Hall effect; Expression for Hall coefficient and its applications. Magnetism: Origin of magnetism; Dia, Para, Ferro, Ferri and Anti-ferro magnetic materials; Hysteresis; Soft and Hard magnetic materials. Applications (qualitative only): Magnetic data storage (MRAM), Hard Disk Drive and Magnetic shielding.

AI Integration	AI-Assisted Hall Effect Analysis for Semiconductor Characterization. AI-Assisted Magnetic Data Storage and MRAM Optimisation. Free/Open-Source AI Tools: Google Colab, Python and Scikit-learn.

Unit– V :	**Low Temperature Physics and Superconductivity**
Content	6. Introduction: Low temperature and its importance; Properties of materials at low temperatures; Liquefaction of gases (basic concept only); Liquid nitrogen and liquid helium (basic properties and uses). Applications (basic ideas): MRI, Rocket fuels, Cryogenic Cooling of High-Performance Computing System. 7. Superconductors: Definitions; Meissner effect; Type I and Type II superconductors; BCS theory: Cooper pairs and energy gap (qualitative); DC and AC Josephson effect. Applications (qualitative only): SQUIDs, magnetic levitation and lossless power transmission. Superconducting qubits: Charge, Flux, Phase.
AI Integration	AI-Assisted Cryogenic Cooling Optimization for High-Performance Computing Systems. AI-Assisted Stability Control in Superconducting Magnetic Levitation Systems. Free/Open-Source AI Tools: Google Colab, Python and Scikit-learn.

Course Outcomes (COs):

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

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

Bloom's Taxonomy Coverage – Summary

Bloom's Level	**Level Name**	**CO(s)**	**Sample Activities**
L3	Apply	**CO1, CO3, CO5**	Interference and diffraction problems; quantum tunnelling application exercises; superconductivity application case studies.

L4	Analyze	**CO2, CO4**	Laser and fiber optic system analysis; Hall effect and magnetic material characterization exercises.

CO – PO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

1. Pandey, B. K., & Chaturvedi, S., *Engineering Physics*, Cengage Learning, 2021.
2. Pillai, S. O., *Solid State Physics*, New Age International, Year not specified.
3. Nielsen, M. A., & Chuang, I. L., *Quantum Computation and Quantum Information*, Cambridge University Press, Year not specified.
4. Saleh, B. E. A., & Teich, M. C., *Fundamentals of Photonics*, Wiley, Year not specified.
5. Keiser, G., *Optical Fiber Communications*, McGraw Hill, Year not specified.
6. Tinkham, M., *Introduction to Superconductivity*, Dover Publications, Year not specified.

Online Resources :

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

5. IBM Qiskit (free, open-source) – For quantum computing and superconducting qubit simulations on Google Colab

Recommended Free & Open-Source AI/Computational Tools :

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
3	0		0	4

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

Prerequisites: Linear Algebra and Calculus

Course Objectives :

The objectives of the course is to

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

Unit– I :	**First-Order Ordinary Differential Equations**
Content	Classification of ODEs by order, degree, and linearity; linear first-order ODEs and integrating factor method; Bernoulli's equation; exact equations and integrating factors. Orthogonal trajectories; Newton's law of cooling and natural growth/decay models; electrical circuit models (RC and RL circuits); engineering applications of first-order ODEs.
AI Integration	Python SymPy dsolve for symbolic ODE solution and verification; direction field (slope field) visualization using Matplotlib; AI-assisted debugging of integrating factor derivations; using ChatGPT/Copilot to explain solution families and physical interpretations; SciPy solve_ivp for numerical ODE solution. Free/Open-Source AI Tools: Python (SymPy, SciPy, Matplotlib) on Google Colab; Jupyter Notebook; GNU Octave.

Unit– II :	**Higher-Order Linear Differential Equations**
Content	Superposition principle; Wronskian and linear independence; complementary function for constant-coefficient ODEs (distinct real, repeated, and complex roots); particular integral by operator method; variation of parameters.

	Simultaneous linear ODEs; Cauchy's (Euler) equation; Legendre's equation; applications to LCR circuits, simple harmonic motion, damped and forced oscillations; mass-spring-dashpot systems.
AI Integration	Python SciPy signal.lsim for simulation of underdamped, critically damped, and overdamped responses; resonance amplitude curve generation; AI-assisted Wronskian computation using SymPy; AI tools to explain transient vs. steady-state response. Free/Open-Source AI Tools: Python (NumPy, SciPy, SymPy, Matplotlib); GNU Octave; WolframAlpha free tier.

Unit– III :	**Partial Differential Equations**
Content	Introduction and classification of PDEs; formation by elimination of arbitrary constants; formation by elimination of arbitrary functions; first-order linear PDEs and Lagrange's method (auxiliary equations); homogeneous linear PDEs with constant coefficients (CF and PI). Standard forms of nonlinear first-order PDEs; second-order PDE applications , wave equation, heat equation, and Laplace's equation; classification as hyperbolic, parabolic, and elliptic.
AI Integration	Python SymPy for PDE formation and symbolic verification; 3D surface plot visualization of PDE solution families using Matplotlib; SciPy finite-difference simulation of the 1D heat equation with animation; AI-assisted PDE classification using the discriminant. Free/Open-Source AI Tools: Python (SymPy, NumPy, SciPy, Matplotlib); WolframAlpha free tier; Scilab.

Unit– IV :	**Vector Differentiation**
Content	Scalar and vector fields; del (∇) operator; gradient and directional derivative; Unitnormal to surfaces; angle between surfaces; divergence and physical interpretation; curl and physical interpretation. Vector identities; conservative fields and scalar potentials; Laplacian; engineering applications of gradient, divergence, and curl in electromagnetic and fluid mechanics contexts.
AI Integration	Python NumPy and Matplotlib for 2D and 3D gradient, divergence, and curl field visualization (quiver plots, level-surface overlays); SymPy for symbolic verification of vector identities; AI tools to interpret Maxwell's equations in terms of divergence and curl operators. Free/Open-Source AI Tools: Python (NumPy, SymPy, Matplotlib, mpl_toolkits); SciPy for numerical computation.

Unit– V :	**Vector Integration and Integral Theorems**
Content	Line integrals over scalar and vector fields; work done and path independence; surface integrals and flux; Green's theorem in the plane and its applications.

	Stokes' theorem; Divergence theorem (Gauss's theorem); applications to area and volume computation; engineering applications in electromagnetism, fluid mechanics, and mass conservation; relationship between integral theorems and Maxwell's equations.
AI Integration	Python SciPy integrate for numerical computation of line and surface integrals; 3D flux visualization with colour-coded normal flux using Matplotlib; numerical verification of Green's and Stokes' theorems; AI-assisted physical interpretation of the Divergence theorem. Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib); SymPy for symbolic integral verification; WolframAlpha free tier.

Bloom's Taxonomy Coverage – Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L3	Apply	**CO1, CO3**	First-order ODE solution problems; PDE formulation and Lagrange's method exercises.
L4	Analyze	**CO2, CO4**	Higher-order ODE response analysis for LCR circuits; gradient/divergence/curl field interpretation.
L5	Evaluate	**CO5**	Green's/Stokes'/Divergence theorem application for flux and circulation computation.

Course Outcomes (COs) :

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

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

CO – PO Articulation Matrix

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

Web Resources & NPTEL Links

1. NPTEL: Ordinary Differential Equations and Applications – https://nptel.ac.in

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

Recommended Free & Open-Source AI/Computational Tools

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
2	0	2	0	3

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

Course Objectives :

The objectives of the course is to

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

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

Unit– II :	**Stacks And Queues**
Content	Stacks: LIFO principle, push, pop, peek; implementation using Python lists; applications , expression evaluation (infix to postfix, postfix evaluation), parenthesis checking. Queues: FIFO principle, enqueue, dequeue; types , Circular Queue, Priority Queue, Deque; implementation using Python lists and linked structures. Applications of queues: BFS simulation, scheduling, real-world analogies. **Source:** TB1: Chapters 4–5 (Vasudevan et al.); TB2: Chapter 6 (Goodrich et al.)

AI Integration	Use VisuAlgo to animate stack and queue operations; use ChatGPT to generate edge cases (overflow, underflow, circular behaviour); use GitHub Copilot to implement expression evaluation.

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

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

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

	applications; use GitHub Copilot to implement hash tables with collision resolution.

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

Course Outcomes (Cos) :

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

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

CO – PO Articulation Matrix :

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

Recommended Free & Open-Source Data Structures / AI Tools

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
1	0	4	0	3

(26ME0301) DESIGN THINKING
(Common to all branches)

Prerequisites: None

Course Objectives ;

The objectives of the course is to

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

Unit– I	**Design Thinking**
Content	Design Thinking: Definition, origin (Stanford d.school, IDEO), and relevance to engineering. The 5-Stage Model: Empathise – Define – Ideate – Prototype – Test. Human-centred design principles: Desirability, Feasibility, Viability. Systems Thinking vs. Linear Thinking: Understanding complexity. Case Studies: Successful design thinking applications (IDEO shopping cart, Stanford Mobile Clinic).
AI Integration	AI Tool: ChatGPT/Copilot – use AI to explain design thinking in different domains (healthcare, education, agriculture). Activity: MindMeister (free, open-source alternative: FreeMind) to create a mindmap of the 5-stage process. AI Video Analysis: summarise key DT principles from documentary case studies.

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

AI Integration	AI Tool: Miro (free plan) for digital empathy maps and POV statements. AI Tool: ChatGPT for generating diverse user persona profiles. Activity: interview peers about a campus problem and use AI to analyse transcripts and extract themes. AI Tool: Dovetail (free plan) for affinity mapping of qualitative research data.

Unit–III :	**Ideation Stage**
Content	Creative Thinking Techniques: Brainstorming, Brainwriting, SCAMPER, Random Word Association. Lateral Thinking: Edward de Bono's Six Thinking Hats. Affinity Diagramming: Grouping ideas by themes. Prioritisation Matrix: Impact vs. Effort, Dot Voting. Concept Selection: Morphological charts and Pugh Matrix.
AI Integration	AI Tool: ChatGPT – use 'expand my idea' prompts to generate variations of a concept. AI Tool: IdeaFlip (free)/Miro for digital brainstorming and affinity diagramming. AI Tool: Midjourney (free trial)/DALL-E (free credits) for visual concept sketches. Activity: run an AI-augmented brainstorming session and compare idea quality with pure human brainstorming.

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

Unit– V :	**Testing, Iteration and AI in Design**
Content	User Testing Methods: Think-aloud protocol, A/B testing, Usability testing. Feedback Analysis: Affinity mapping of test feedback.Iteration Cycle: Incorporating feedback and pivoting. Design Presentation: Pitching to stakeholders (elevator pitch, demo day format). AI in Design: Overview of AI-assisted design (generative design, co-creation with AI).
AI Integration	AI Tool: Maze (free plan) – conduct remote usability tests on Figma prototypes. AI Tool: ChatGPT – analyse collected user feedback and generate insights and improvement suggestions. Activity: present final prototype to a

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

Bloom's Taxonomy Coverage – Summary

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

Course Outcomes (COs)

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

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

CO-PO Mapping Matrix :

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

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

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

Text Books :

1. Tim Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*, 1st Edition, Harper Business, 2019.
2. Michael Lewrick, Patrick Link and Larry Leifer, *The Design Thinking Playbook*, 1st Edition, Wiley, 2018.

Reference Books :

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

Web Resources :

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

YouTube / Video Resources :

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	2	1

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

Prerequisites: Programming fundamentals in Python. (Inferred – not specified in the source.)

Course Objectives :

The objectives of the course is to

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

List of Experiments :

MODULE I: Introduction Algorithm Analysis & Linear Structures

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

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

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

Experiments:

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

MODULE II: Stacks And Queues

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

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

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

Experiments:

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

MODULE III: Linked Lists

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

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

Suggested Tools: Python Tutor, VisuAlgo, Jupyter Notebook

Experiments:

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

MODULE IV: Trees And Heaps

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

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

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

Experiments:

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

MODULE V: Graphs And Hashing

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

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

Suggested Tools: Python, NetworkX, Gephi, VisuAlgo

Experiments:

1. **Experiment 14: Graph Representation and Traversal ,** Represent graphs using adjacency matrix and adjacency list; implement BFS and DFS.
2. **Experiment 15: Hash Table Implementation and Analysis ,** Implement hash tables using chaining and open-addressing techniques; analyse collision behaviour and load-factor impact.

Course Outcomes (COs) :

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

CO – PO Articulation Matrix

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

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

Textbooks :

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

Reference Books :

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

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

Online Learning Resources:

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

Recommended Free & Open-Source Data Structures / AI Tools

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26HS0850) PHYSICS FOR ADVANCED COMPUTING LAB
(Common to CSE & Allied branches)

Course Objectives :

The objectives of this course is to

1. *Develop practical skills in conducting physics experiments relevant to advanced computing technologies.*
2. *Understand the physical principles underlying modern electronic, optical, and semiconductor devices used in computing systems.*
3. *Analyze experimental data using computational tools and interpret results through scientific methods.*
4. *Explore the applications of materials, lasers, optical fibers, and nanotechnology in advanced computing and communication.*
5. *Enhance problem-solving, analytical thinking, teamwork, and technical report-writing skills through laboratory experiments.*

List of Experiments :

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

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

Course Outcomes :

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

1. *Perform physics laboratory experiments accurately using standard scientific instruments and follow laboratory safety practices.*
2. *Analyze and interpret experimental data related to semiconductor materials, optical systems, lasers, and advanced materials used in computing technologies.*
3. *Apply computational techniques and software tools for data acquisition, visualization, and analysis of experimental results.*
4. *Explain the role of modern materials, nanotechnology, photonics, and sensors in the design and development of advanced computing systems.*
5. *Prepare professional laboratory reports and effectively communicate experimental observations, results, and conclusions.*

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	2	1

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

Course Objectives :

The objectives of this course is to

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

Lab Syllabus – 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	Speeko, ChatGPT, Google Assistant	**CO1, CO3, CO5**

			for spontaneous speaking		
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, Notebook LM	**CO2, CO4**
10	**Debate & Argumentation Skills**	Structured debate (for/against); Rebuttal techniques; logical fallacy identification	AI generates arguments on both sides; evaluate logical structure using ChatGPT	ChatGPT, Claude.ai, Kialo Edu	**CO2, CO3**

11	**Report Writing & Technical Communication**	Lab report, incident report, and project report formats; headings, data presentation	AI improves report coherence and passive/active voice usage; grammar and structure feedback	Grammarly Business, ChatGPT, Hemingway Editor	**CO4, CO5**
12	**Etiquette, Netiquette & Personal Branding**	Workplace conversation role play; LinkedIn profile creation; professional social media etiquette	AI reviews LinkedIn summaries; digital persona analysis; personal brand statement via ChatGPT	ChatGPT, Canva AI, LinkedIn AI features	**CO3, CO4, CO5**

Course Outcomes :

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

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

Text Books :

1. Raman, Meenakshi, & Sharma, Sangeeta. *Technical Communication: Principles and Practice*. 3rd Edition. 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.

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 :

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

Technical English for Engineers	NPTEL / SWAYAM	IIT Roorkee	https://swayam.gov.in

Lab Software & AI Tools Required :

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

CO – PO / PSO MAPPING

Correlation Level: 3 = High | 2 = Medium | 1 = Low | – = No Correlation

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**3**	**-**	**-**	**2**	**-**	**1**	**2**	**-**	**-**	**-**	**-**	**-**	**3**	**2**
CO2	**2**	**-**	**-**	**2**	**-**	**2**	**1**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
CO3	**2**	**-**	**-**	**3**	**-**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
CO4	**2**	**-**	**-**	**2**	**-**	**2**	**1**	**-**	**-**	**-**	**-**	**-**	**3**	**2**
CO5	**1**	**-**	**-**	**2**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**2**	**3**
Average	**2.0**	**-**	**-**	**2.2**	**3.0**	**2.0**	**1.6**	**-**	**-**	**-**	**-**	**-**	**2.4**	**2.6**

Mapping Legend

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

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

I B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	1	0.5

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

Course Objectives :

The objectives of this course is to

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

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

	• ULO1.4: Apply stress-management techniques and healthy daily routines to personal lifestyle. [Apply] • ULO1.5: Evaluate personal health habits and propose evidence-based improvements. [Evaluate]

Unit-II :	**Yoga, Pranayama, Meditation and Mind-Body Practice**
Topics	• Introduction to Yoga – history, philosophy, Patanjali's Ashtanga Yoga • Surya Namaskar – 12-step sequence with correct breathing • Asanas for flexibility: Tadasana, Trikonasana, Paschimottanasana, Sarvangasana • Asanas for strength & balance: Virabhadrasana I & II, Vrikshasana, Naukasana • Pranayama techniques: Anulom-Vilom, Bhastrika, Kapalbhati, Bhramari • Meditation and mindfulness – Dharana, Dhyana, body-scan, guided relaxation • Yoga Nidra for sleep and stress relief • Yoga for specific conditions – stress, back pain, PCOD/PCOS, anxiety • Shatkarmas (cleansing processes) – an overview; International Day of Yoga
Unit-wise Learning Outcomes (ULOs)	**On completion of this unit, students will be able to:** • ULO2.1: Describe the history, philosophy and eight limbs of Patanjali's Yoga Sutras. [Remember/Understand] • ULO2.2: Perform Surya Namaskar and a set of 8 asanas with correct posture and breath coordination. [Apply] • ULO2.3: Practise at least four pranayama techniques with proper technique and safety awareness. [Apply] • ULO2.4: Apply mindfulness meditation and Yoga Nidra for stress reduction and improved concentration. [Apply] • ULO2.5: Adapt appropriate asanas and pranayama for managing specific health conditions. [Analyse]

Unit-III :	**Physical Fitness, Sports Skills and Sports Technology**
Topics	• Physical fitness components – strength, endurance, flexibility, agility, speed, coordination • Principles of training • – overload, progression, specificity, reversibility, individuality • Warm-up, cool-down and dynamic stretching routines

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

Course Outcomes (COs) ;

CO	Course Outcome	Bloom's Level
CO1	Explain the fundamental concepts of health, wellness, fitness and the dimensions of holistic well-being.	Remember / Understand
CO2	Demonstrate correct yogic practices – asanas, pranayama and meditation – for physical and mental health.	Apply
CO3	Design a personal fitness plan integrating nutrition, exercise, stress management and lifestyle modification.	Analyse / Apply
CO4	Understand rules, skills and safety norms for a chosen sport and participate actively with sportsmanship.	Apply
CO5	Evaluate personal health indicators and reflect on lifestyle choices using technology and evidence-based practices.	Evaluate

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

Text Books :

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

Reference Books :

1. Silva, Mira & Shyam Mehta, *Yoga: The Iyengar Way*, Dorling Kindersley, 2019.
2. Jim Clover, *Sports Medicine Essentials*, Cengage Learning, 2020.
3. Charles A. Bucher, *Foundations of Physical Education & Sport*, McGraw-Hill Education, 2018.
4. Mark Williams & Danny Penman, *Mindfulness: An Eight-Week Plan for Finding Peace*, Rodale Books, 2011.

Web Resources :

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

YouTube Resources

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

AI-Integrated Activities :

S.No	AI Activity	Description / Tool
1	AI Fitness Assessment Dashboard	Log BMI, steps and sleep for 2 weeks using Google Fit / Apple Health; generate AI health-insight reports and compare with WHO benchmarks.
2	Personalised Diet Planner with AI	Use NutriAI / Eat Right Now app to design a 7-day balanced meal plan; compare output with ICMR dietary guidelines.

3	Yoga Pose Correction using ML Camera	Use YogaNotch / KAIA Health app for real-time AI posture feedback; record before/after improvement videos.
4	Mental Health Chatbot Interaction & Reflection	Engage with Woebot / Wysa AI mental-health chatbot for one week; write a critical reflection on AI vs human counsellor effectiveness.
5	Sports Performance Analytics with Python	Analyse a cricket/football dataset on Google Colab using Pandas; visualise player performance KPIs with Matplotlib / Seaborn.

CO – PO / PSO Mapping

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

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
CO1	2	-	-	-	-	3	2	-	-	-	-	-	-	-
CO2	-	-	3	-	-	3	3	-	-	-	-	-	-	-
CO3	-	2	3	3	-	3	3	-	-	-	-	-	2	-
CO4	-	-	3	-	-	2	2	-	3	-	-	-	-	-
CO5	-	3	-	3	2	3	2	-	-	-	-	-	2	-

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

I B.Tech – II Sem.

L	T	Pr	P	C
2	0	0	0	0

(26CS4801) CYBERSECURITY AWARENESS FOR ENGINEERS
(CSE)

Prerequisites: None

Course Objectives :

The objectives of this course is to

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

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

Unit– II :	**Cyber Threats, Cybercrime and Social Engineering**
Content	Cyber threats and vulnerabilities; Cybercrime and cyber offenders; Malware: Virus, Worm, Trojan, Spyware, Ransomware; Cyberattack lifecycle and common attack methods.Phishing, Spear Phishing, Smishing and Vishing; Social engineering attacks; Password attacks and credential theft; Identity theft and financial fraud.

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

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

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

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

Course Outcomes (COs) :

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

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

CO – PO Articulation Matrix

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

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning	PSO1 Computing Core	PSO2 Profnl Dev.
CO1	L2	2	1	–	–	1	2	–	2	–	1	–	2	1	2
CO2	L2	2	2	–	1	1	2	–	2	–	1	–	2	1	2
CO3	L3	2	2	1	1	2	2	–	2	1	1	–	3	2	2
CO4	L2	1	1	–	–	–	3	1	3	1	2	–	2	1	2
CO5	L3	2	2	1	1	2	2	1	2	1	1	1	3	2	3

Textbooks :

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

Reference Books :

1. Charles J. Brooks, Christopher Grow, Philip Craig and Donald Short, *Cybersecurity Essentials*, 1st Edition, Wiley, 2018.
2. William Stallings and Lawrie Brown, *Computer Security: Principles and Practice*, 4th Edition, Pearson, 2018.
3. Michael E. Whitman and Herbert J. Mattord, *Principles of Information Security*, 6th Edition, Cengage Learning, 2018.
4. Joseph Migga Kizza, *Guide to Computer Network Security*, 2nd Edition, Springer, 2015.
5. Eric Cole, *Cybersecurity for Beginners*, 1st Edition, Wiley, 2023.

Online Resources :

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

Recommended Cybersecurity Awareness Resources & Tools :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
2	0	2	0	3

(26HS0869) PROBABILITY AND STATISTICS
(Common to all branches)

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

Course Objectives :

The objectives of this course is to

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

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

Unit-II :	Probability and Random Variables
Content	Sample space and events; axioms of probability; addition and multiplication theorems; conditional probability; Bayes' theorem. Random variables: discrete and continuous; probability mass function; probability density function; cumulative distribution function; mathematical expectation and variance.
AI Integration	Python simulation of probability experiments using NumPy random module; visualization of PMF, PDF, and CDF using Matplotlib; AI-assisted derivation and verification of Bayes' theorem problems; simulation of conditional

	probability scenarios using SciPy. Free/Open-Source AI Tools: Python (NumPy, SciPy, Matplotlib) on Google Colab; SymPy; WolframAlpha free tier.

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

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

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

Bloom's Taxonomy Coverage , Summary

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

Course Outcomes (COs) :

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

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

CO-PO Articulation Matrix

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

CO	BL	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11
CO1	L1	3	1	,	,	,	,	,	,	,	,	,
CO2	L2	3	2	,	1	,	,	,	,	,	,	,

CO3	L3	3	2	1	1	1	,	,	,	,	,	,
CO4	L4	3	3	1	2	1	,	,	,	,	,	,
CO5	L5	3	3	2	3	1	1	,	,	1	1	,

CO-PO Weightage :

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

Web Resources & NPTEL Links :

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

6. Statsmodels Documentation: https://www.statsmodels.org
7. WolframAlpha (free tier) for verification: https://www.wolframalpha.com

Recommended Free & Open-Source AI/Computational Tools :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

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

(Common to all branches)

Prerequisites: None

Course Objectives :

The objectives of the course is

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

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

Unit-II :	**Harmony in the Individual**
Content	**Harmony between Self and Body:** Needs of the Self (happiness, trust, etc.) vs. Needs of the Body (food, shelter, clothing); Four Dimensions: Icchha (desire), Vichar (thought), Anubhav (experience), Bodh (realisation).

	Feelings in Relationships: Nine feelings , Trust, Respect, Affection, Care, Guidance, Reverence, Glory, Gratitude, Love; Understanding Happiness and Prosperity. **Self-regulation vs. External Regulation:** Inner motivation and its role in ethical conduct.
AI Integration	**AI Tool:** Replika (free AI companion) for exploring emotional needs conversations and reflecting on AI limitations. Activity: Well-being Wheel using Canva; AI-assisted gap identification between material and value-based happiness.

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

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

Unit-V :	**Ethical Human Conduct and Professional Ethics**
Content	Right Understanding, Right Feeling, Right Action: The value-ethics-skill triad; Professional Ethics: Codes of conduct for engineers (ASME, IEEE, IEI).

	Corruption, Dishonesty, and Short-term Thinking: Causes and value-based remedies; Vision for a Value-based Society; AI Ethics: Bias, fairness, transparency, and accountability.
AI Integration	**AI Tool:** IBM AI Fairness 360 to demonstrate AI bias using sample datasets. **Case Study:** Bhopal Gas Tragedy Challenger Disaster through professional ethics lens. **Capstone:** Personal Value Charter drafted with ChatGPT assistance.

Bloom's Taxonomy Coverage , Summary

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

Course Outcomes (COs) :

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

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

CO , PO PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

YouTube Video Resources :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
2	0	0	0	2

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

Prerequisites: None

Course Objectives :

The objectives of the course is to

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

Unit-I :	**Introduction to Managerial Economics**
Content	**Managerial Economics:** Meaning, Nature, Scope and significance for Engineers; Economic Analysis , Micro vs. Macroeconomics; positive vs. normative economics. **Demand Analysis:** Concept, Determinants, Demand Function, Law of Demand; Elasticity of Demand: Types, Measurement, Business Applications; Demand Forecasting: Meaning, Need, Methods.
AI Integration	**AI Tool:** ChatGPT to generate demand scenarios for an engineering product and interpret price elasticity. **AI Tool:** Google Sheets + FORECAST function for Moving Average and Exponential Smoothing. **Activity:** AI-assisted regression demand forecasting in Google Colab.

Unit-II :	**Production and Cost Analysis**
Content	Production Concepts and Modern Manufacturing Systems; **Production Function:** Isoquants and Isocosts; Least-Cost Combination of Inputs. **Cost Concepts:** Fixed, Variable, Total, Average, Marginal Costs; **Break-Even Analysis:** Concept, Assumptions, Applications; Margin of Safety and Profit Planning.

AI Integration	**AI Tool:** Google Sheets/LibreOffice Calc for BEA spreadsheet with dynamic charts. **AI Tool:** ChatGPT to generate cost functions and identify optimal production level. **Activity:** AI-assisted CVP analysis using Plotly for cost curve visualisation.

Unit-III :	**Business Organisations and Market Structures**
Content	**Forms of Business Organisations:** Sole Proprietary, Partnership, Joint Stock Companies, Startups and Digital Enterprises; Public vs Private Sector in Digital Economy. **Market Structures:** Classification with real-world examples; Pricing Methods and Strategies.
AI Integration	**AI Tool:** ChatGPT to simulate a duopoly pricing game and find Nash Equilibrium. **Activity:** analyse pricing strategies of competing tech products. **Discussion:** AI-driven dynamic pricing in e-commerce (Amazon, Uber Surge Pricing).

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

Unit-V :	**Financial Accounting and Analysis**
Content	Accounting Principles and Digital Accounting Systems; Journal, Ledger, Trial Balance; Final Accounts: Trading Account, Profit and Loss Account, Balance Sheet with Simple Adjustments. Financial Analysis: Liquidity Ratios, Activity Ratios, Capital Structure Ratios, Profitability Ratios.
AI Integration	Automated financial statement preparation using Python (pandas). **AI Tool:** ChatGPT to interpret company financial ratios. **Project:** ratio analysis of an Indian engineering company's annual report (BHEL, L&T, Infosys).

Bloom's Taxonomy Coverage , Summary

Bloom's Level	**Level Name**	**CO(s)**	**Sample Activities**

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

Course Outcomes (COs) :

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

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

CO , PO PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

AI Tools Used In This Course :

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

AI-Integrated Activities :

Unit	Activity	Description	AI Tool Used	Outcome Mapped
I	Demand Forecasting Simulation	Use AI tools to forecast demand for a consumer product using historical data	ChatGPT + Statista	CO1
I	Elasticity Chatbot Quiz	AI-generated adaptive quiz on types and measurement of elasticity	Google Gemini	CO1
II	Break-Even Calculator	Build a Break-Even model using Excel + Copilot for sensitivity analysis	Microsoft Copilot	CO2
II	Cost Curve Visualisation	Plot and interpret cost curves using AI-assisted data tools	Wolfram Alpha Tableau	CO2
III	Market Structure Case Study	Generate real-world case studies of oligopoly (e.g., Telecom sector) using AI	Claude AI ChatGPT	CO3
III	Startup Evaluation Report	Evaluate a digital startup's business model using AI-generated insights	Google Gemini	CO3

IV	Capital Budgeting Decision Tool	Use AI to compare NPV, IRR for two competing engineering projects	Copilot + Wolfram Alpha	CO4
IV	IRR Sensitivity Analysis	Explore how changes in cash flows affect IRR using AI-powered spreadsheets	Microsoft Copilot	CO4
V	Digital Accounting Exercise	Prepare journal, ledger, and trial balance using Tally AI Zoho Books	Tally AI Zoho AI	CO5
V	Financial Ratio Dashboard	Construct a financial ratio dashboard from a company's annual report using Tableau AI	Tableau AI	CO5

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

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0505) DIGITAL LOGIC AND COMPUTER ORGANIZATION
(CSE)

Prerequisites: Computational Thinking and Problem Solving with Python(co-requisite)

Course Objectives :

The objectives of the course is to

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

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

Unit-II :	Digital Logic Circuits , II and Basic Structure of Computers
Content	Sequential Circuits, Flip-Flops (SR, JK, D, T), master-slave flip-flops, flip-flop conversions, Binary counters, Registers, Shift Registers (SISO, SIPO, PISO, PIPO), Ripple counters, Synchronous counters. Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance, Multiprocessors and multicomputers, Computer Generations, Von-Neumann Architecture.
AI Integration	Use an AI/LLM tool to explain flip-flop truth tables, excitation tables, and conversion methods (e.g. JK to D). Use a Python template in Google Colab to simulate a 4-bit synchronous up-counter and SIPO shift register, verifying state sequences. Use an AI/LLM tool to explain Von-Neumann architecture, fetch-decode-execute cycle, and compare with Harvard architecture.

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

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

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

Bloom's Taxonomy Coverage , Summary

Bloom's Level	Level Name	CO(s)	Sample Activities
L2/L3	Understand/Apply	**CO4, CO5**	Memory hierarchy and cache mapping explanation; I/O method comparison and bus structure description.
L3	Apply	**CO1, CO3**	K-map simplification problems; Booth's algorithm and IEEE 754 floating-point exercises.
L4	Analyze	**CO2**	Sequential circuit design and flip-flop conversion analysis; counter and shift register behaviour evaluation.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply number systems, binary codes, Boolean algebra, and logic gate principles to represent, convert, and simplify digital logic functions using K-maps.	**L3 (Apply)**
CO2	Design combinational circuits , adders, comparators, multiplexers, and decoders , and analyze sequential circuits including flip-flops, counters, and shift registers.	**L4 (Analyze)**

CO3	Apply computer arithmetic techniques , 2's complement arithmetic, carry-lookahead addition, Booth's multiplication, and IEEE 754 floating-point operations.	**L3 (Apply)**
CO4	Explain memory organization including semiconductor memories, cache memory, virtual memory, and programmable logic devices (PLA, PAL, FPGA).	**L2,L3**
CO5	Describe I/O organization methods , programmed I/O, interrupt-driven I/O, and DMA , and explain bus structures, interface circuits, and standard I/O interfaces (USB, PCI).	**L2,L3**

CO , PO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

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

II B.Tech , II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0506) ALGORITHMS DESIGN AND ANALYSIS
(Professional Core-II)
(Common to CSE ,CSM,CAD)

Course Objectives :

The objectives of the course is to

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

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

Unit-II :	Divide And Conquer
Content	**Divide and conquer paradigm:** strategy, recurrence formulation, subproblem independence. Sorting: Merge Sort (algorithm, analysis, stable), Quick Sort (partitioning schemes, randomized Quick Sort, expected complexity), Heap Sort review.

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

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

Unit-IV :	**Dynamic Programming**
Content	**DP principles:** optimal substructure, overlapping subproblems; memoization (top-down) vs. tabulation (bottom-up); reconstructing solutions from DP tables. **Classic DP problems:** Longest Common Subsequence (LCS) , table construction and traceback; 0/1 Knapsack , DP table filling; Edit Distance (Levenshtein) , string alignment; Optimal Binary Search Tree. **Advanced DP:** Floyd-Warshall all-pairs shortest path , $O(V^3)$, negative edge handling, detecting negative cycles; Travelling Salesman Problem (TSP) , Held-Karp DP (exact); Subset Sum.

	Source: TB1: Chapters 15, 25 (Cormen et al.); TB2: Chapter 6 (Kleinberg & Tardos)
AI Integration	Use DP Visualizer and Python Tutor to animate LCS table and knapsack DP matrix; compare memoization vs. tabulation on execution time and memory; use ChatGPT to verify step-by-step DP trace for edit distance.

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

Bloom's Taxonomy Coverage , Summary

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

Course Outcomes (COs

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

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

CO , PO PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0507) OBJECT ORIENTED DESIGN AND PROGRAMMING
(Professional Core -III)
(Common to CSE ,CSM,CAD)

Course Objectives :

The objectives of the course is to

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

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

Unit-II :	**Classes, Objects, And Methods**
Content	**Classes and objects:** class declaration, modifiers, class members, object creation, assigning objects; access control (public, private, protected, default);

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

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

Unit-IV :	**Packages, Exception Handling, And Java I/O**
Content	**Packages:** defining and importing packages, class path, access control; Java.lang , Object class, Enumeration, Math, Wrapper classes, auto-boxing/unboxing; java.util , Formatter, Random, Time package (Instant, TemporalAdjusters); date/time formatting. **Exception handling:** hierarchy of exception classes; keywords throws, throw, try, catch, finally; multiple catch clauses; class Throwable; checked vs. unchecked exceptions; try-with-resources. **Java I/O:** standard I/O streams; byte streams, character streams; Scanner class; reading and writing files using Java NIO (Files class).

	Source: TB1: Chapters 12,14 (Seth & Juneja); TB3: Chapter 15 (Deitel & Deitel) , Java I/O and Files
AI Integration	Use draw.io to design multi-package system architecture; use ChatGPT to generate programs with exceptions, identify errors, and fix them; use GitHub Copilot for file I/O programs; use AI to explore Java API (Math, Random, Time).

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

Course Outcomes (Cos) :

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

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

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

CO , PO PSO Articulation Matrix (NBA)

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

CO	Bloom's Lvl	PO1 Eng. Know.	PO2 Problem Analysis	PO3 Design Dev.	PO4 Invest-igation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sust.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgm
CO1	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	**2**	,
CO2	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	1	,
CO3	L3	**3**	**3**	**2**	1	**2**	,	,	,	,	1	,
CO4	L3	**3**	**3**	**2**	1	**2**	,	,	,	,	1	,
CO5	L3	**3**	**2**	**2**	1	**2**	,	,	,	,	1	,
CO6	L4	**3**	**3**	**3**	**2**	**2**	,	,	,	1	1	,

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
0	0	0	2	1

(26CS0508) DIGITAL LOGIC AND COMPUTER ORGANIZATION LAB

(CSE)

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

Course objectives :

The objectives of this course is

1. *To provide hands-on experience in number systems, Boolean algebra, K-map simplification, and combinational logic circuit design.*
2. *To develop practical skills in designing and verifying sequential circuits using digital logic simulators.*
3. *To enable simulation and analysis of computer arithmetic operations and processor architecture concepts.*
4. *To familiarize students with cache memory organization and I/O transfer techniques through simulation tools***.**

List of Experiments :

Unit-I :	Number Systems, Logic Gates and K-Map Simplification

1. Number System Conversion and Boolean Expression Verification

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

2. K-Map Simplification and Logic Gate Verification

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

3. Universal Gate Implementation , Decoder and Multiplexer Design

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

Unit-II :	Sequential Circuits and Computer Architecture Simulation

1. Flip-Flop Truth Table Verification and Conversion

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

2. Synchronous Counter and Shift Register Simulation

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

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

Trace through a simplified fetch-decode-execute cycle simulation for a sample instruction set, identifying the role of each functional unit (ALU, Control Unit, Memory, I/O). Compare Von-Neumann and Harvard architecture data flow paths and document differences in a summary table.

Unit-III :	**Computer Arithmetic, Memory, and I/O Simulation**

1. Carry-Lookahead Adder Design and Performance Comparison

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

2. Booth's Algorithm Multiplication Simulation

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

3. IEEE 754 Floating-Point Representation and Addition

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

4. Cache Memory Mapping Simulation

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

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

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

AI Tools Integration (Lab)

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

Course Outcomes (Cos) :

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

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

CO , PO Articulation Matrix (NBA)

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

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

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

Textbooks :

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

Reference Books :

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

Online Resources :

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

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

II B.Tech , II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0509) ALGORITHMS DESIGN AND ANALYSIS LAB
(Professional Core -II Lab)
(Common to CSE ,CSM,CAD)

Prerequisites: Data Structures, Programming for Problem Solving (Python/C/Java). (Inferred – not stated in source.)

Course Objectives :

The objectives of the course is to

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

List of Experiments :

MODULE I: Algorithm Analysis And Mathematical Foundations

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

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

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

Experiments:

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

- **Experiment-3: Recursion Tree Visualization** Implement Merge Sort and Binary Search, trace their recursive execution, compare runtime, and generate recursion trees using VisuAlgo and Python Tutor.
- **Experiment-4: AVL Tree and B Tree Analysis** Implement AVL tree insertion with rotations and B Tree insertion, compare balancing operations, and use VisuAlgo and ChatGPT to generate insertion sequences and visualize balancing.

MODULE II: Divide And Conquer

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

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

Suggested Tools: VisuAlgo; Algorithm Visualizer; ChatGPT

Experiments:

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

MODULE III: Greedy Algorithms

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

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

Suggested Tools: ChatGPT; VisuAlgo; Gephi; Cytoscape

Experiments:

- **Experiment-8: Activity Selection and Scheduling** Implement Activity Selection, Fractional Knapsack, and Job Scheduling, using AI tools to generate scheduling scenarios and compare optimal schedules.
- **Experiment-9: Minimum Spanning Tree and Shortest Path** Implement Prim's Algorithm, Kruskal's Algorithm, and Dijkstra's Algorithm, using Gephi and Cytoscape alongside VisuAlgo to visualize graph traversal and compare path selection.

- **Experiment-10: Huffman Encoding** Generate character frequencies, construct a Huffman Tree, and calculate the compression ratio, using AI tools to generate frequency datasets and compare compression.

MODULE IV: Dynamic Programming

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

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

Suggested Tools: DP Visualizer; Python Tutor; ChatGPT

Experiments:

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

MODULE V: Advanced Topics And Computational Complexity

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

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

Suggested Tools: Graphviz; Gephi; Claude; ChatGPT

Experiments:

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

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Apply asymptotic notation, amortized analysis, and recurrence solving techniques to analyse algorithm complexity.	L3 (Apply)
CO2	Design and analyse divide-and-conquer algorithms for sorting, searching, and matrix multiplication.	L4 (Analyse)
CO3	Apply greedy algorithms to graph and optimization problems.	L3 (Apply)
CO4	Design dynamic programming solutions for classical optimization problems.	L4 (Analyse)
CO5	Apply complexity classification, backtracking, branch-and-bound, and approximation methods to computationally difficult problems.	L4 (Analyse)

CO , PO Articulation Matrix

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

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

Textbooks :

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

Reference Books :

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

Online Learning Resources :

1. NPTEL , Design and Analysis of Algorithms , nptel.ac.in
2. MIT OpenCourseWare , Introduction to Algorithms , ocw.mit.edu
3. VisuAlgo , visualgo.net
4. Algorithm Visualizer , algorithm-visualizer.org

5. Gephi , gephi.org
6. Python Tutor , pythontutor.com
7. Wolfram Alpha , wolframalpha.com

Recommended Free & Open-Source AI Computational Tools :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0510) OBJECT ORIENTED DESIGN AND PROGRAMMING LAB
(Professional Core -III Lab)
(Common to CSE ,CSM,CAD)

Prerequisites: Programming for Problem Solving; Data Structures.

Course Objectives :

The objectives of the course is to

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

List of Experiments :

MODULE I: Oop Fundamentals, Data Types, Operators & Control Structures

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

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

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

Experiments:

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

MODULE II: Classes, Objects, Methods & Collections

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

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

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

Experiments:

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

MODULE III: Arrays, Inheritance & Interfaces

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

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

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

Experiments:

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

MODULE IV: Packages, Exception Handling & Java I/O

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

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

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

Experiments:

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

MODULE V: Strings, Multithreading, Jdbc & Javafx

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

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

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

Experiments:

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

Course Outcomes (COs) :

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

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

CO , PO Articulation Matrix

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

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

Textbooks :

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

Reference Books :

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

Online Learning Resources :

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

Recommended Free & Open-Source Java AI Tools :

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

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

II B.Tech , I Sem.

L	T	Pr	P	C
1	0	0	2	2

(26CS0570) AI ASSISTED APPLICATION DEVELOPMENT
(Skill Enhancement Course- I)
(Common to CSE & Allied branches)

Prerequisites: Basic programming knowledge in any language. (Inferred – not specified in the source.)

Course Objectives :

The objectives of the course is to

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

List of Experiments :

MODULE I: Frontend Development With Html & Css (Weeks 1,4)

Topics: HTML5 semantic elements , text, images, hyperlinks, lists, tables, multimedia; responsive web forms and validation interfaces; CSS selectors, typography, colors, Flexbox, Grid, and the CSS Box Model; reusable UI components , navigation bars, cards, dashboards, responsive menus.

AI Integration: HTML generation and UI refinement; form design and responsive styling; CSS generation and layout optimization; component generation and UI enhancement.

Suggested Tools: HTML5, CSS3, VS Code

Experiments:

1. **Week 1: AI-Assisted Web Page Design using HTML ,** Develop a personal portfolio website using HTML5 semantic elements including text, images, hyperlinks, lists, tables, and multimedia; use AI tools to generate and refine webpage structure and improve layout.

2. **Week 2: Responsive Web Forms using HTML and CSS ,** Design responsive registration and login forms with appropriate input controls and a validation interface; generate and optimize form layout using AI assistance.
3. **Week 3: CSS Styling and Responsive Layout Development ,** Develop responsive webpages using CSS selectors, typography, colors, Flexbox, Grid, and the CSS Box Model; compare manually designed and AI-generated layouts.
4. **Week 4: User Interface Component Development ,** Create reusable UI components including navigation bars, cards, dashboards, and responsive menus; use AI tools to improve UI consistency and responsiveness.

MODULE II: Client-Side Programming With Javascript (Weeks 5,8)

Topics: JavaScript ES6 fundamentals , variables, operators, loops, functions, arrays, objects; DOM manipulation and event handling; client-side form validation; browser storage (Local Storage, Session Storage); API integration and dynamic content using the Fetch API and JSON.

AI Integration: Code generation and debugging; event-logic generation; validation generation and code correction; API integration support and response handling.

Suggested Tools: JavaScript ES6, Browser Console, REST APIs

Experiments:

1. **Week 5: JavaScript Programming Fundamentals ,** Develop interactive web pages using variables, operators, loops, functions, arrays, and objects; use AI assistance to generate and optimize scripts.
2. **Week 6: DOM Manipulation and Event Handling ,** Develop event-driven applications using DOM operations, event listeners, dynamic content updates, and user-interaction handling.
3. **Week 7: Client-side Form Validation and Browser Storage ,** Implement form validation using JavaScript and store user information using Local Storage and Session Storage.
4. **Week 8: API Integration and Dynamic Content Generation ,** Consume external APIs and dynamically update webpage content using the Fetch API and JSON processing.

MODULE III: Backend Development With Node.Js & Express (Weeks 9,11)

Topics: Backend server setup with routing and middleware; RESTful API development with CRUD operations; database integration with MongoDB; user authentication and authorization.

AI Integration: Server generation and debugging; API generation and testing; query generation and schema optimization.

Suggested Tools: Node.js, Express.js, MongoDB, Postman

Experiments:

1. **Week 9: Backend Application Setup using Node.js and Express.js ,** Create a backend server application with routing and middleware support; generate boilerplate backend code using AI tools.

2. **Week 10: REST API Development and CRUD Operations ,** Develop RESTful APIs supporting Create, Read, Update, and Delete operations.
3. **Week 11: Database Integration and Authentication ,** Connect backend applications with MongoDB and implement user authentication and authorization mechanisms.

MODULE IV: Full-Stack Integration & Deployment (Weeks 12,14)

Topics: Frontend,backend integration; complete application workflows; full-stack testing and performance optimization; documentation; cloud deployment and application monitoring.

AI Integration: Integration support and issue resolution; test generation and optimization; deployment assistance and troubleshooting.

Suggested Tools: React.js, Node.js, REST APIs, GitHub

Experiments:

1. **Week 12: Frontend,Backend Integration ,** Integrate frontend interfaces with backend APIs and implement complete application workflows.
2. **Week 13: Full-Stack Application Testing and Optimization ,** Test application functionality, improve performance, and generate documentation using AI assistance.
3. **Week 14: Deployment and Application Monitoring ,** Deploy applications to cloud platforms and monitor application behavior and performance.

MODULE V: Ai-Assisted Web Application Mini Project (Week 15)

Topics: End-to-end MERN-stack application development integrating frontend, backend, database, testing, deployment, and documentation.

AI Integration: Code generation, debugging, testing, documentation, and deployment support.

Suggested Tools: MERN Stack (MongoDB, Express.js, React.js, Node.js), GitHub

Experiments:

1. **Week 15: AI-Assisted Web Application Mini Project ,** Develop and demonstrate a complete web application integrating frontend, backend, database, testing, deployment, and documentation.

Suggested Mini Projects:

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

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Design and develop responsive web interfaces using HTML and CSS with AI-assisted development tools.	L6 (Create)
CO2	Develop interactive client-side applications using JavaScript and AI-assisted coding and debugging techniques.	L6 (Create)
CO3	Develop server-side applications and integrate databases for web application development.	L6 (Create)
CO4	Build, test, and integrate full-stack web applications using AI-assisted development workflows.	L6 (Create)
CO5	Deploy and maintain web applications while applying responsible and effective use of AI tools in software development.	L5 (Evaluate)

CO , PO Articulation Matrix

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

CO	Bloom's Level	PO1 Eng. Knowledge	PO2 Problem Analysis	PO3 Design/ Dev.	PO4 Investi-gation	PO5 Modern Tools	PO6 Engineer & Society	PO7 Environ. & Sustain.	PO8 Ethics	PO9 Team Work	PO10 Commun.	PO11 Prj Mgmt & Finance	PO12 Lifelong Learning	PSO1 Computing Core	PSO2 Profnl Dev.
CO1	L6	3	2	3	2	3	,	,	,	1	2	1	2	3	2
CO2	L6	3	3	3	2	3	,	,	,	1	2	1	2	3	2
CO3	L6	3	3	3	2	3	,	,	,	1	1	2	2	3	2
CO4	L6	3	3	3	3	3	,	,	,	2	2	2	3	3	3
CO5	L5	3	2	3	2	3	1	1	2	2	3	3	3	3	3

Textbooks :

1. Duckett, J., *HTML and CSS: Design and Build Websites*, Wiley Publications.
2. Flanagan, D., *JavaScript: The Definitive Guide*, O'Reilly Media.
3. Subramanian, V., *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React and Node*, Apress.

Reference Books :

1. Sebesta, R. W., *Programming the World Wide Web*, Pearson Education.
2. Brown, E., *Web Development with Node and Express*, O'Reilly Media.
3. Ng, A., *Generative AI for Everyone*, DeepLearning.AI.
4. Johnson, R. A., and Bhattacharyya, G. K., *Statistics: Principles and Methods*, 7th Edition, Wiley, 2014.

Online Learning Resources :

1. MDN Web Docs , HTML, CSS & JavaScript Documentation , developer.mozilla.org
2. W3Schools , Web Development Tutorials , w3schools.com
3. React Documentation , react.dev

4. Node.js Documentation , nodejs.org/docs
5. Express.js Documentation , expressjs.com
6. MongoDB Documentation , mongodb.com/docs
7. GitHub Skills & GitHub Education , skills.github.com
8. Visual Studio Code Documentation , code.visualstudio.com/docs
9. OpenAI Developer Resources , platform.openai.com/docs
10. freeCodeCamp Web Development , freecodecamp.org
11. https://www.youtube.com/@JoyofLearningCSE/featured

Recommended Free & Open-Source AI Web Development Tools :

Frontend: HTML5, CSS3, React.js , responsive interfaces and reusable components.
Backend & Database: Node.js, Express.js, MongoDB , server-side logic and data storage.
Tooling: VS Code, Postman, GitHub , editing, API testing and version control.
AI Assistants: ChatGPT, GitHub Copilot , code generation, debugging, testing and documentation.

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

II B.Tech , I Sem.

L	T	Pr	P	C
2	0	0	0	0

(26HS0812) ENVIRONMENTAL SCIENCE
(Common to all branches)

Prerequisites: No specific prerequisites

Course Objectives :

The objectives of the course is to

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

Unit-I :	**Multidisciplinary Nature of Environmental Studies and Natural Resources**
Core Content	**Multidisciplinary Nature of Environmental Studies: ,** Definition, Scope and Importance , Need for Public Awareness. **Natural Resources:** Renewable and non-renewable resources , Natural resources and associated problems , Forest resources , Use and over-exploitation, deforestation, case studies , Timber extraction , Mining, dams and other effects on forest and tribal people , Water resources , Use and over utilization of surface and ground water , Floods, drought, conflicts over water, dams , benefits and problems , Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies , Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies , Energy resources: Use and exploitation of conventional and non-conventional energy sources, case studies.

AI Integration	AI in Natural Resource Monitoring: Remote sensing and satellite imagery analysis using Google Earth Engine (free cloud platform) for real-time deforestation detection and land use change mapping. Machine learning for groundwater level prediction from historical well data (Python/scikit-learn). AI-based precision agriculture , reducing fertilizer and pesticide use through crop health monitoring using drone imagery and computer vision (overview). ML forecasting of renewable energy output (solar irradiance, wind speed) for energy resource planning. Free Open-Source AI Tools: Google Earth Engine (free cloud GIS); QGIS (free, open-source); Python (scikit-learn, Matplotlib) , Google Colab (free); Global Forest Watch (free dashboard).

Unit-II :	**Ecosystems and Biodiversity and its Conservation**
Core Content	**Ecosystems:** Concept of an ecosystem , Structure and function of an ecosystem , Producers, consumers and decomposers , Energy flow in the ecosystem , Ecological succession , Food chains, food webs and ecological pyramids , Introduction, types, characteristic features, structure and function of: Forest ecosystem; Grassland ecosystem; Desert ecosystem; Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries). **Biodiversity and its Conservation:** Introduction , Definition: genetic, species and ecosystem diversity , Bio-geographical classification of India , Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values , Biodiversity at global, national and local levels , India as a mega-diversity nation , Hot-spots of biodiversity , Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts , Endangered and endemic species of India , Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

AI Integration	AI in Ecosystem and Biodiversity Studies: AI-powered species identification applications such as iNaturalist (free) use image recognition to identify plants, animals and insects from photographs, enabling citizen science biodiversity surveys. ML for biodiversity hotspot mapping using satellite imagery and species distribution models (MaxEnt , free). Computer vision for automated wildlife camera trap image analysis , identifying and counting species without manual review. Google Earth Engine for ecosystem health monitoring: NDVI time-series analysis for vegetation cover change detection. AI-assisted assessment of in-situ vs. ex-situ conservation effectiveness. Free Open-Source AI Tools: iNaturalist (free app/web); MaxEnt (free species distribution model); Google Earth Engine (free); QGIS (free); Python (rasterio, geopandas) , Google Colab.

Unit-III :	**Environmental Pollution and Solid Waste Management**
Core Content	**Environmental Pollution:** Definition, cause, effects and control measures of: Air Pollution; Water Pollution; Soil Pollution; Marine Pollution; Noise Pollution; Thermal Pollution; Nuclear Hazards. **Solid Waste Management:** Causes, effects and control measures of urban and industrial wastes , Role of an individual in prevention of pollution , Pollution case studies , Disaster Management: floods, earthquake, cyclone and landslides.
AI Integration	AI in Pollution Monitoring and Disaster Management: ML-based Air Quality Index (AQI) prediction from meteorological data and traffic density (Python/scikit-learn/TensorFlow , free). IoT sensor networks with AI-driven anomaly detection for real-time water quality monitoring. Computer vision for automated solid waste sorting and classification (overview of AI-enabled sorting robots). OpenAQ platform (free, open data) for accessing global air quality sensor data for ML model training. AI and GIS (QGIS) for disaster risk zone mapping , flood inundation modelling, earthquake vulnerability assessment and cyclone track prediction. Free Open-Source AI Tools: OpenAQ (free global AQ data API); QGIS (free); Python (scikit-learn, TensorFlow, Matplotlib) , Google Colab (free); CPCB data portal (free).

Unit-IV :	Social Issues and the Environment
Core Content	**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).

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Taxonomy Level
CO1	Recall and explain the multidisciplinary nature of environmental studies, define key concepts of scope and public awareness, and describe the uses, over-exploitation and associated environmental problems of forest, water, mineral, food and energy resources with reference to Indian and global case studies.	L1,L2 (Remember, Understand)
CO2	Describe the structure and function of different ecosystem types, explain ecological succession, food chains and ecological pyramids, and outline the concept and classification of biodiversity, its value, threats and conservation strategies (in-situ and ex-situ) including India's status as a mega-diversity nation.	L2,L3 (Understand, Apply)

CO3	Identify causes, effects and control measures of air, water, soil, marine, noise, thermal and nuclear pollution; apply principles of solid waste management for urban and industrial wastes; analyze pollution case studies and evaluate disaster management strategies for floods, earthquakes, cyclones and landslides.	L3,L4 (Apply, Analyze)
CO4	Analyze social issues related to unsustainable development; evaluate the impacts of climate change, global warming, acid rain and ozone layer depletion; examine Indian environmental legislation (EPA, Air Act, Water Act, Wildlife Protection Act, Forest Conservation Act) and assess their enforcement challenges.	L4,L5 (Analyze, Evaluate)
CO5	Describe population growth trends and their environmental impacts; identify links between environment and human health; demonstrate the role of information technology and AI-based tools in environmental monitoring, public health surveillance and field data collection; apply digital tools (QGIS, Python, iNaturalist) for environmental documentation.	L2,L3 (Understand, Apply)

CO , PO PSO Articulation Matrix (NBA)

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

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

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

Textbooks :

1. Bharucha, E., *Textbook of Environmental Studies for Undergraduate Courses*, Universities Press.
2. Palaniswamy, *Environmental Studies*, Pearson Education.
3. Unnisa, S. A., *Environmental Studies*, Academic Publishing Company.

4. Nambiar, K. R., *Text Book of Environmental Studies for Undergraduate Courses as per UGC Model Syllabus*, Scitech Publications (India) Pvt. Ltd.

Reference Books :

1. Dave, D., and Reddy, E. S. B., *Textbook of Environmental Science*, Cengage Publications.
2. Reddy, M. A., *Text Book of Environmental Sciences and Technology*, BS Publications.
3. Sharma, J. P., *Comprehensive Environmental Studies*, Laxmi Publications.
4. Henry, J. G., and Heinke, G. W., *Environmental Sciences and Engineering*, Prentice Hall of India Private Limited.
5. Chatwal, G. R., *A Text Book of Environmental Studies*, Himalaya Publishing House.
6. Masters, G. M., and Ela, W. P., *Introduction to Environmental Engineering and Science*, Prentice Hall of India Private Limited.

Web Resources : & Online Platforms :

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
Google Earth Engine (free cloud platform) , real-time satellite data analysis for deforestation, land use and vegetation change detection
QGIS (free, open-source GIS) , environmental mapping, watershed delineation, disaster risk zone mapping, biodiversity hotspot analysis
Python (NumPy, Pandas, Matplotlib, scikit-learn, TensorFlow) on Google Colab (free) , AQI prediction, population growth ML models, climate data analysis
iNaturalist (free, open-source citizen science) , species identification using image recognition during field work; community biodiversity database
eBird (free, Cornell Lab) , bird species recording and mapping during field visits; global bird distribution data
OpenAQ (free, open-source) , global real-time and historical air quality data for ML model training

MaxEnt (free) , species distribution modelling for biodiversity hotspot prediction from occurrence data
Global Forest Watch (free dashboard) , monitoring deforestation and forest cover change using satellite imagery
CMIP6 climate datasets (free, ESGF portal) , global climate model data for Python-based climate change scenario analysis

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

I B.Tech - I Sem.

L	T	Pr	P	C
2	0	0	0	2

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

Course Outline :

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

Course Objectives :

The objectives of this course is

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

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

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

Unit-III :	Human Systems and Bio-Designs
Content	Brain as CPU (EEG, robotic prosthetics), Eye as camera (optical corrections, lens materials).
AI Integration	Heart as pump (ECG, stents),Kidney as filtration (dialysis systems).
Free/Open-Source AI Tools:	AI-based ECG anomaly detection, Robotic arm design inspired by neural signals, Use of AI imaging tools to detect cataracts, Simulate stent design using ANSYS.

Unit-IV :	Bio-Inspired Materials and Mechanisms
Content	Photosynthesis → photovoltaic cells,Bird flight → GPS and aircraft navigation.
AI Integration	Lotus leaf → superhydrophobic surfaces,Kingfisher beak → bullet train design.
Free/Open-Source AI Tools:	Model solar energy conversion inspired by photosynthesis, AI simulation of lotus-leaf hydrophobicity, Kingfisher-inspired aerodynamic modelling, Use of AI to design self-cleaning surfaces.

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

Course Outcomes :

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

CO	Course Outcome Statement
CO1	Explain the fundamentals of biology, biomolecules, and human systems relevant to engineering applications.
CO2	Apply AI tools and computational techniques for biological data analysis, biomolecule visualization, and healthcare applications.
CO3	Analyze biomolecules and bio-inspired systems for solving engineering problems and developing sustainable technologies.
CO4	Design biomimetic and bioengineering solutions using interdisciplinary engineering approaches and AI-enabled tools.

CO5	Evaluate emerging trends in bioengineering such as 3D bioprinting, regenerative medicine, and AI-based diagnostics for societal application

CO – PO Articulation Matrix

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

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12
CO1	3	2	1	1	1	–	–	–	–	–	–	2
CO2	2	3	2	2	3	–	–	–	1	–	–	2
CO3	2	3	3	2	2	2	3	–	–	–	–	2
CO4	2	2	3	2	3	2	2	1	2	1	–	2
CO5	2	2	2	2	3	3	2	1	1	1	–	3

Text Books :

1. Rajendra Singh C., Rathnakar Rao N., *Biology for Engineers*, Rajendra Singh C. and Rathnakar Rao N. Publishing, Bengaluru, 2023.
2. Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., Jaganthan M.K., *Biology for Engineers*, Tata McGraw-Hill, New Delhi, 2012.

Reference Books :

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

Web links and Video Lectures (e-Resources):

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

7. https://www.classcentral.com/subject/biology
8. https://www.futurelearn.com/courses/biology-basic-concepts

AI Tools for the Course:

1. Deep Chem – drug discovery simulations.
2. Alpha Fold – protein structure prediction.
3. Generative Design AI – biomaterial innovation.

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

II B.Tech – II Sem.

L	T	Pr	P	C
2	0	0	0	2

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

Course Objectives :

The objectives of this course is

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

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

Unit-II :	**Sustainable Materials and Resource Selection**
Content	Sustainable materials and resource efficiency; Indoor air quality considerations; Recycled and alternative material sources; Durability and service life assessment. Life cycle of engineering materials; Multi-criteria decision analysis for material selection.

AI Integration	ML for sustainable material selection using multi-criteria decision analysis in Python. AI-based prediction of material performance using regression models (scikit-learn). Comparative LCA using openLCA. Free/Open-Source AI Tools: scikit-learn; openLCA; Python (NumPy, scikit-learn) on Google Colab.

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

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

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

Bloom's Taxonomy Coverage – Summary

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

Course Outcomes (COs)

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

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

CO – PO / PSO Articulation Matrix :

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

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	**2**	1	–	–	–	**3**	**3**	**2**	–	**2**	–	**2**	**2**	**2**
CO2	**2**	**2**	**2**	**2**	**2**	**2**	**3**	**2**	1	**2**	–	**2**	**2**	**2**
CO3	**2**	**2**	**2**	**2**	**3**	**2**	**3**	1	1	**2**	1	**2**	**2**	**3**
CO4	**2**	**2**	**3**	**2**	**3**	**3**	**3**	**2**	1	**2**	1	**3**	**2**	**3**
CO5	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**2**	**2**	**3**	**2**	**3**	**2**	**3**
CO-PO Correlation Highlights	CO4 and CO5 show strong correlation with PO3 (Design) and PO5 (Modern Tools) through green rating compliance and AI-driven sustainable design. All COs map strongly to PO7 (Environment &													

	Sustainability), reflecting the course's core focus. CO5 additionally connects to PO10 and PO12 through integrative project-based sustainable solution design.

Textbooks :

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

Reference Books :

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

Online Resources :

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

Web Resources & Nptel Links :

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

Recommended Free & Open-Source AI/Computational Tools

1. **openLCA (free, open-source)** – Life Cycle Assessment software for embodied carbon calculation
2. EnergyPlus (free, US DOE) – Building energy simulation for operational energy analysis
3. OpenStudio (free, NREL) – Building energy modeling interface for EnergyPlus

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

II B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0511) OPEARTING SYSTEMS AND INTELLIGENT RESOURCE MANAGEMENT
(Professional Core – IV)
(Common to CSE,IOT,CSD branches)

Prerequisites: Computer Organization.

Course Objectives:

The objectives of the course is to

1. *Understand the structure and functions of operating systems.*
2. *Analyze process management, scheduling, and concurrency mechanisms.*
3. *Apply memory management, file systems, and deadlock handling techniques.*
4. *Explore AI-based approaches for resource management in operating systems.*
5. *Understand modern operating system trends including cloud* and container systems.

Unit-I :	**Operating Systems Fundamentals, System Structures & Process Management**
Content	Operating Systems Overview: definition and role of an operating system; functions of OS process management, memory management, file systems, I/O systems; operating system operations and services; types of operating systems batch, time-sharing, real-time, distributed, mobile; computing environments traditional, mobile, distributed, cloud; open-source operating systems and their significance. System Structures: OS services and system programs; user interface CLI and GUI; system calls concept and types (process control, file management, device management, information maintenance, communication); system programs and utilities; OS design and implementation approaches; OS structures — monolithic, layered, modular; system boot process; OS debugging techniques. Process Management: process concept and lifecycle; Process Control Block (PCB); process scheduling concepts; scheduling criteria (CPU utilization, throughput, turnaround time, waiting time, response time); scheduling algorithms — FCFS, SJF, Priority, Round Robin; multiprocessor scheduling; client–server communication models.
AI Integration	Use ChatGPT or Claude to simulate process scheduling step-by-step and system-call workflows; use GitHub Copilot to implement scheduling algorithms in Python; use AI-based diagram tools to visualize process lifecycle and OS architecture.

Suggested Tools:	ChatGPT, Claude, GitHub Copilot, Python

Unit-II :	**Operating Systems Fundamentals, System Structures & Process Management**
Content	Concurrency & Multithreading: process vs. thread; multithreading models many-to-one, one-to-one, many-to-many; thread libraries and threading issues; inter-process communication (IPC) shared memory, message passing. Synchronization: race conditions and the critical-section problem; synchronization solutions mutex locks, semaphores, monitors; busy waiting and blocking; classical synchronization problems Dining Philosophers, Readers–Writers. Memory Management: memory hierarchy and allocation; swapping; contiguous memory allocation; paging and segmentation. Virtual Memory: concept of virtual memory; demand paging; copy-on-write; page replacement algorithms FIFO, LRU, Optimal; frame allocation strategies; thrashing; memory-mapped files; kernel memory allocation.
AI Integration	Use Google Colab and Python to simulate page replacement algorithms; use ChatGPT to debug synchronization problems and generate solutions for concurrency issues; use Python Tutor and visualizers to visualize thread execution and memory allocation.
Suggested Tools:	Google Colab, Python, ChatGPT, Python Tutor

Unit-III :	**Deadlocks, File Systems, Storage & Security**
Content	Deadlocks: system model and resource allocation; deadlock conditions mutual exclusion, hold and wait, no preemption, circular wait; deadlock prevention; deadlock avoidance (Banker's Algorithm); deadlock detection and recovery. File Systems: file concepts and attributes; file operations; directory structures single-level, two-level, tree; file system implementation allocation methods, directory implementation. Secondary Storage: disk structure and disk attachment; disk scheduling algorithms FCFS, SSTF, SCAN, C-SCAN. Protection & Security: goals and principles of protection; access matrix and access control mechanisms; revocation of access rights; system security program threats (virus, worms, trojans), system and network threats, cryptography basics, user authentication methods, firewalls and intrusion detection.
AI Integration	Use Linux tools for file-system and permission handling; use Wireshark and log tools to analyze system/network behavior; use ChatGPT to simulate deadlock scenarios and generate security case studies.

Suggested Tools:	Linux, Wireshark, ChatGPT

Unit-IV :	**Intelligent Operating Systems & AI-Based Resource Management**
Content	Introduction to AI in operating systems; intelligent CPU scheduling using machine learning; predictive resource allocation techniques; reinforcement learning for scheduling decisions; adaptive memory management; workload prediction and performance optimization; self-healing and self-optimizing operating systems; AI-driven cloud resource orchestration.
AI Integration	Use Scikit-learn to build workload prediction models; use Google Colab to train and test ML models; use ChatGPT to generate datasets and explain model behavior.
Suggested Tools:	Scikit-learn, Google Colab, ChatGPT

Unit-V :	**Modern Operating Systems, Virtualization & Cloud Systems**
Content	Evolution of modern operating systems; virtualization concepts and hypervisors; containerization Docker architecture and usage; container orchestration Kubernetes basics; cloud operating systems and service models; edge and distributed operating systems; microkernel vs. monolithic vs. hybrid architectures; security in modern OS environments.
AI Integration	Use Docker for container creation and deployment; use Kubernetes for basic orchestration and scheduling; use ChatGPT to generate Dockerfiles and troubleshoot deployment.
Suggested Tools:	Docker, Kubernetes, ChatGPT

Course Outcomes :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Explain OS structure, services, and system calls.	L2 (Understand)
CO2	Analyze process scheduling and concurrency mechanisms.	L4 (Analyze)
CO3	Apply memory management, file systems, and deadlock handling techniques.	L3 (Apply)

CO4	Evaluate AI-based approaches for intelligent resource management.	L5 (Evaluate)
CO5	Analyze modern OS architectures including virtualization and cloud systems.	L4 (Analyze)

CO – PO Articulation Matrix

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

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

Textbooks :

1. Silberschatz, P. B. Galvin & G. Gagne, *Operating System Concepts*, 10th ed., Wiley, 2018.
2. S. Tanenbaum & H. Bos, *Modern Operating Systems*, 4th ed., Pearson, 2015.

Reference books :

1. D. M. Dhamdhere, *Operating Systems: A Concept-Based Approach*, 3rd ed., McGraw-Hill, 2017.
2. S. Tanenbaum & A. S. Woodhull, *Operating Systems: Design and Implementation*, 3rd ed., Pearson, 2006.
3. S. Tanenbaum & M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed., Pearson, 2017.
4. T. Erl, R. Puttini & Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, 2013.

Web Resources

1. NPTEL – Operating Systems – nptel.ac.in/courses/106106144
2. Docker Documentation – docs.docker.com
3. Kubernetes Documentation – kubernetes.io
4. Linux Foundation Training – training.linuxfoundation.org
5. Coursera – Operating Systems and Systems Programming – coursera.org
6. https://www.youtube.com/@JoyofLearningCSE/featured

Recommended Free & Open-Source OS / AI Tools

1. **Operating System:** Linux (Ubuntu) – process, memory and file-system experimentation.
2. **Programming & ML:** Python, Scikit-learn – algorithm implementation and workload prediction.
3. **Containers & Cloud:** Docker, Kubernetes – virtualization and orchestration.

4. **Analysis:** Wireshark, Python Tutor – system/network and execution visualization.
5. **AI Assistants:** ChatGPT, Claude, GitHub Copilot – simulation, debugging and Dockerfile generation.
6. **Notebooks:** Google Colab – ML model training and testing.

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

II B.Tech – II Sem.

L	T	Pr	P	C
3	0	0	0	3

(26CS0512) INTELLIGENT DATABASE MANAGEMENT SYSTEMS
(Professional Core -V)
(Common to CSE ,CSM,CAD branches)

Prerequisites: Data Structures; basic programming in Python.

Course Objectives :

The objectives of the course is to

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

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

Unit-II :	Relational Model and SQL
Content	Relational model: relations, tuples, attributes, domains, schemas; relational algebra – selection, projection, union, set difference, Cartesian product, natural join, outer joins, intersection. SQL: DDL (CREATE, ALTER, DROP); DML (INSERT, UPDATE, DELETE); DQL (SELECT with WHERE, ORDER BY, GROUP BY, HAVING, aggregate functions, joins, subqueries, set operations); DCL (GRANT, REVOKE); TCL (COMMIT, ROLLBACK, SAVEPOINT). Views, indexes, and query optimization: simple and complex views; B-Tree and B+ Tree indexing (dense vs. sparse); primary and secondary indexes; query optimization basics – cost-based optimizer, predicate pushdown. *(Source: TB1 Ch.2–6, 11; TB2 Ch.5–6, 14–15)*
AI Integration	Use MySQL and SQLite for SQL query implementation; use ChatGPT to generate SQL from natural language and verify manually; use GitHub Copilot to scaffold complex queries; use Google Colab for SQL exercises.
Suggested Tools:	MySQL; SQLite; ChatGPT; GitHub Copilot; Google Colab

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

Unit-IV :	Transaction Management and Concurrency Control

Content	Transaction concepts: ACID properties (Atomicity, Consistency, Isolation, Durability); transaction states; COMMIT, ROLLBACK, SAVEPOINT; isolation levels – READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE. Concurrency control: lock-based protocols (shared and exclusive locks, two-phase locking, strict 2PL); timestamp-based protocols; deadlock detection (waits-for graph), prevention (wound-wait, wait-die), avoidance. Serializability: conflict serializability; precedence-graph construction; view serializability. Recovery: log-based recovery (undo and redo); Write-Ahead Log (WAL); checkpointing; ARIES algorithm (Analysis, Redo, Undo). *(Source: TB1 Ch.14–17; TB2 Ch.19–21)*
AI Integration	Use Python simulations to create concurrent transaction schedules; use ChatGPT to analyse serializability and trace precedence-graph construction; simulate lost-update and dirty-read scenarios and resolve them.
Suggested Tools:	Python; ChatGPT

Unit-V :	**Intelligent Data Management for AI Systems**
Content	NoSQL systems: motivation (3Vs, schema flexibility, horizontal scale); CAP theorem; document stores (MongoDB), key-value (Redis), column-family (Cassandra), graph (Neo4j – Cypher). Text-to-SQL and LLM-over-database: natural-language-to-SQL translation; schema linking; accuracy challenges; RAG-over-SQL; LLM as data analyst; verification requirements before production use. Vector data management: vector databases (FAISS, Pinecone, Weaviate, Chroma); ANN search (HNSW, IVF); similarity metrics (cosine, Euclidean, dot product); hybrid search; re-ranking. Data governance: data catalog (Apache Atlas), lineage tracking; GDPR Article 5; India DPDP Act 2023. *(Source: TB1 Ch.10, 22–23; TB2 Ch.24–25; Kleppmann)*
AI Integration	**AI Integration:** Use ChatGPT to generate SQL from natural language and verify each query; use Claude to compare HNSW vs. IVF for a vector-search workload; use MongoDB Atlas for document-store experiments; use an AI assistant to generate a data-governance checklist.
Suggested Tools:	ChatGPT; Claude; MongoDB Atlas; FAISS; Pinecone

Course Outcomes

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

CO	Course Outcome Statement	Bloom's Level
CO1	Explain DBMS architecture, data models, and ER modelling; design ER diagrams and map them to relational schemas.	L4 (Analyze)
CO2	Apply relational algebra and write SQL queries for data definition, manipulation, retrieval, and control.	L3 (Apply)
CO3	Apply normalization techniques (1NF through BCNF) to eliminate redundancy and design efficient relational schemas.	L4 (Analyze)
CO4	Analyse transaction schedules for serializability; apply concurrency control protocols and recovery mechanisms.	L4 (Analyze)
CO5	Evaluate modern database systems including NoSQL and vector databases; apply AI-assisted querying and data governance.	L5 (Evaluate)

CO – PO Articulation Matrix

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

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

Text Books :

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

Reference Books :

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

Online resources :

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

Recommended Free &Free-Tier Ai/Database Tools

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

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

II B.Tech – II Sem.

L	T	Pr	P	C
2	0	2	0	3

(26CS0513) AUTOMATA THEORY AND COMPILER DESIGN
(Professional Core – VI)
(CSE)

Prerequisites: Discrete Mathematics and basic programming in C / Python. (Inferred – not specified in the source.)

Course Objectives :

The objectives of this course is to

1. *Build a strong foundation in formal languages, automata, and the mathematical limits of computation.*
2. *Analyze and construct finite automata, grammars, and Turing machines.*
3. *Understand compiler design phases from lexical analysis to code generation.*
4. *Design and evaluate parsing techniques and syntax-directed translation.*
5. *Apply optimization techniques and understand target code generation.*

Unit-I :	**Introduction to Finite Automata**
Content	**Core Content:** Introduction to Finite Automata: Alphabets, Strings, Languages. Finite Automata, Nondeterministic Finite Automata (NFA), Finite Automata with Epsilon-Transitions. Deterministic Finite Automata (DFA), String and Language Acceptance, Conversion of NFA with ε-transitions to NFA without ε-transitions, Conversion of NFA to DFA, DFA minimization, equivalence of DFA.
Ai Integration	Stepwise DFA construction using ChatGPT; automata visualization explanations using Gemini; designing DFA and NFA for specific string acceptance; converting NFA with ε-transitions to NFA and NFA to DFA; DFA minimization and checking equivalence between machines.
Suggested Tools:	JFLAP, Python

Unit-II :	**Regular Expressions & Context-Free Grammars**
Content	Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular

	Expressions, Conversion of Finite Automata to Regular Expressions. Pumping Lemma for Regular Languages — statement of the pumping lemma, applications of the pumping lemma. Context-Free Grammars: definition of context-free grammars, derivations using a grammar, leftmost and rightmost derivations, the language of a grammar, parse trees, ambiguity in grammars and languages. Chomsky Normal Form, Greibach Normal Form.
AI Integration	Regex to DFA conversion; developing CFGs and parse trees while resolving ambiguity; grammar normalization (CNF, GNF); regex optimization via Claude AI; grammar correction using Gemini; parse tree generation.
Suggested Tools:	JFLAP, draw.io

Unit-III :	**Pushdown Automata & Turing Machines**
Content	**Core Content:** Pushdown Automata: definition of the pushdown automaton, the languages of a PDA, equivalence of PDAs and CFGs, acceptance by final state and empty stack. Turing Machines: introduction to Turing machine, formal description, instantaneous description, the language of a Turing machine, types of Turing machine — Multi-Tape Turing Machine, Non-Deterministic Turing Machine.
AI Integration	Designing PDAs for acceptance by final state and empty stack; constructing Turing Machines and tracking Instantaneous Descriptions; building Multi-Tape and Non-Deterministic TM models; TM design logic and transition assistance via Claude AI; logic verification for complex machine models using ChatGPT.
Suggested Tools:	JFLAP, draw.io

Unit-IV :	**Introduction to Compiler Design and Parsing**
Content	**Core Content:** Introduction to Compiler Design: the structure of a compiler. Lexical Analysis: the role of the lexical analyzer, input buffering, recognition of tokens. Syntax Analysis: introduction; Top-Down Parsing: Recursive Descent and LL(1); Bottom-Up Parsing: Shift-Reduce and Operator Precedence.
AI Integration	Implementing lexical analyzers for token recognition and input buffering; constructing LL(1) parsing tables and Recursive Descent parsers; applying Shift-Reduce and Operator Precedence bottom-up parsing; code generation

	for lexical analyzers using GitHub Copilot; parser debugging and error analysis via Perplexity AI.
Suggested Tools:	Flex / Bison, VS Code

Unit-V :	**LR Parsing, Code Generation & Optimization**
Content	**Core Content:** Introduction to LR Parsing: Simple LR, CLR, LALR. Semantic Analysis: Syntax-Directed Translation Schemes. Intermediate-Code Generation: Three-Address Code. Code Optimization and Generation: principal sources of optimization, basic blocks and flow graphs, optimization of basic blocks, issues in the design of a code generator, the target language, a simple code generator, peephole optimization.
AI Integration	Optimization suggestions via Claude AI; code translation and research via NotebookLM; designing parsing tables for SLR, CLR, and LALR; generating Three-Address Code (TAC) and Syntax-Directed Translation; constructing flow graphs and applying peephole optimization.
Suggested Tools:	GCC, LLVM, Graphviz

Course Outcomes (COs) :

CO	Course Outcome Statement	Bloom's Level
CO1	Explain formal language hierarchy and automata models.	L2 (Understand)
CO2	Construct and minimize DFA/NFA and derive regular expressions.	L3 (Apply)
CO3	Analyze context-free grammars and design parsers (LL/LR).	L4 (Analyze)
CO4	Apply syntax-directed translation and semantic analysis.	L3 (Apply)
CO5	Generate and optimize intermediate and target code.	L4 (Analyze)

CO – PO Articulation Matrix :

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

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

Text Books :

1. Michael Sipser, *Introduction to the Theory of Computation*, 3rd ed., Cengage Learning.
2. K. L. P. Mishra & N. Chandrasekaran, *Automata Theory and Formal Languages*, Revised ed., PHI Learning, 2022.
3. John C. Martin, *Theory of Computation*, 2021 ed., McGraw-Hill.
4. A. V. Aho, M. S. Lam, R. Sethi & J. D. Ullman, *Compilers: Principles, Techniques and Tools*, 2nd ed., Pearson.
5. K. V. SUnitha & N. Kalyani, *Compiler Design*, McGraw-Hill, 2021/2022.

Web Resources :

1. NPTEL – Compiler Design & Theory of Computation – nptel.ac.in
2. Stanford University – CS theory & compiler lecture materials – web.stanford.edu
3. GeeksforGeeks – Automata theory & compiler design tutorials – geeksforgeeks.org
4. LLVM Foundation – LLVM compiler infrastructure documentation – llvm.org/docs

Recommended Tools & AI Assistants :

Operating System: Linux (Ubuntu).
Programming: Python, C.
Debugging: GDB, VS Code Debugger.
Visualization: JFLAP, Graphviz, draw.io.
Compiler Tools: Flex / Bison, GCC, LLVM.
Containerization: Docker.
AI Tools: ChatGPT, Claude, Gemini, Perplexity AI, GitHub Copilot, NotebookLM.

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

II B.Tech – II Sem.

L	T	Pr	P	C
2	0	2	0	3

(26CS0514) SOFTWARE ENGINEERING FOR AI DRIVEN SYSTEM

(Professional Core – VII)
(Common to CSE,CSD branches)

Prerequisites: Programming fundamentals and basic problem-solving skills. (Inferred – not specified in the source.)

Course Objectives :

The objectives of this course is to

1. *Understand software engineering principles, process models, and software lifecycle activities.*
2. *Analyze software requirements and apply project management techniques for software development.*
3. *Apply software design principles and agile engineering practices for building quality software systems.*
4. *Evaluate software quality through testing methodologies and software engineering automation practices.*
5. *Apply AI-assisted tools and techniques throughout the software development lifecycle.*
6. *Understand software engineering practices for AI-enabled systems and emerging development environments.*

Unit-I :	**Software Engineering Foundations And Process Models**
Content	Software Engineering Fundamentals: evolution of software systems, software crisis, exploratory style of development, software development practices, computer system engineering principles. Software Process Models: waterfall model and extensions, incremental model, Rapid Application Development (RAD), spiral model, agile development model; comparative analysis of process models. Software life cycle concepts; software development projects. Emerging trends: human–AI collaboration in software development; AI-assisted software lifecycle overview.
AI Integration	Use ChatGPT or Claude to generate software project scenarios and compare SDLC models; use Gamma.app or Notion AI to create process model explanations and lifecycle visualizations; use Miro or Lucidchart AI to visualize software process workflows.

Suggested Tools:	ChatGPT, Claude, Gamma.app, Notion AI, Miro, Lucidchart AI, GitHub Copilot

Unit-II :	**Software Project Management And Requirements Engineering**
Content	Software Project Management: complexities, responsibilities of project managers, metrics for project size estimation, project estimation techniques, empirical estimation techniques, COCOMO, Halstead software science, risk management. Requirements Engineering: requirements gathering, requirements analysis, Software Requirements Specification (SRS). Formal specification methods: axiomatic specification, algebraic specification, executable specification; Fourth Generation Languages (4GL). AI-Enabled Requirements Engineering: automated requirement extraction, AI-assisted documentation generation, requirement consistency analysis.
AI Integration	Use ChatGPT or Claude to generate and refine SRS documents; use Notion AI for requirement documentation and organization; use Jira AI or Linear for project planning and task management; use Claude to validate requirements and identify risks.
Suggested Tools:	ChatGPT, Claude, Notion AI, Jira AI, Linear

Unit-III :	**Software Design And Agile Engineering Practices**
Content	Software Design Principles: characteristics of good design, layered arrangement of modules, cohesion and coupling, software design approaches. Function-Oriented Software Design: Structured Analysis and Design, Data Flow Diagrams (DFD), structured design, detailed design, design review. Agile Engineering Practices: agility and cost of change, agile process models, Extreme Programming (XP), agile toolsets. User Interface Design: characteristics of user interfaces, GUI concepts, component-based GUI development, user interface methodology. Modern Development Approaches: component-based development, low-code / no-code application development, comparison with traditional development.
AI Integration	Use StarUML or Visual Paradigm AI for UML modeling; use Figma AI or Uizard for user interface prototyping; use Mendix or OutSystems for low-code development; use AppGyver or Bubble for no-code development; use Miro AI or draw.io for DFD and workflow creation.

Suggested Tools:	StarUML, Visual Paradigm AI, Figma AI, Uizard, Mendix, OutSystems, Bubble, AppGyver, Miro AI, draw.io

Unit-IV :	**Software Testing, Quality Engineering, And Automation**
Content	Software Testing: coding and code review, software documentation, black-box testing, white-box testing, debugging, program analysis tools, integration testing, testing object-oriented programs, smoke testing. Software Quality Engineering: software reliability, statistical testing, quality management systems — ISO 9000, Capability Maturity Model (CMM), Six Sigma. Software Engineering Automation Practices: automated test generation, continuous testing concepts, automated code review, static code analysis, AI-assisted defect prediction.
AI Integration	Use GitHub Copilot for code generation and AI-assisted code review; use Testim or Selenium AI for automated test case generation and testing; use SonarQube for static code analysis; use DeepCode or Snyk for AI-based defect identification.
Suggested Tools:	GitHub Copilot, Claude, Testim, Selenium AI, SonarQube, DeepCode, Snyk, ChatGPT

Unit-V :	**Software Engineering For Ai-Driven Systems**
Content	AI System Engineering Fundamentals: characteristics of AI systems, traditional software vs. AI systems, data-centric development. AI Lifecycle Management: data pipelines, feature engineering, model lifecycle, model validation, model deployment. AI-Supported Software Maintenance: predictive maintenance using AI, AI-assisted bug localization, reverse engineering using AI, AI-generated documentation, model drift and retraining. Responsible AI Engineering: explainable AI (XAI) concepts, bias and fairness, ethical considerations, AI governance, monitoring AI systems.
AI Integration	Use Google Colab or Jupyter Notebook for AI workflow experimentation; use MLflow for model tracking and lifecycle management; use Evidently AI or Grafana for model monitoring and drift visualization; use Hugging Face Hub to explore pretrained models; use Claude or Notion AI for AI-generated documentation and maintenance support.
Suggested Tools:	Google Colab, Jupyter Notebook, MLflow, Evidently AI, Grafana, Hugging Face Hub, Claude, Notion AI

Course Outcomes (COs)

CO	Course Outcome Statement	Bloom's Level
CO1	Explain software engineering principles, lifecycle models, and software development practices.	L2 (Understand)
CO2	Analyze software requirements and apply project estimation and management techniques.	L4 (Analyze)
CO3	Apply software design methodologies and agile practices to software systems.	L3 (Apply)
CO4	Evaluate software quality using testing, reliability, and software engineering automation practices.	L5 (Evaluate)
CO5	Design software solutions using AI-enabled tools and practices for modern software systems.	L6 (Create)

CO – PO Articulation Matrix :

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

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

Text Books :

1. R. S. Pressman & B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill Education, 2020.
2. I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2016.
3. C. Huyen, *AI Engineering: Building Applications with Foundation Models*, O'Reilly Media, 2025.

Reference Books :

1. C. Huyen, *Designing Machine Learning Systems*, O'Reilly Media, 2022.
2. V. Lakshmanan, S. Robinson & M. Munn, *Machine Learning Design Patterns*, O'Reilly Media, 2020.
3. P. Jalote, *An Integrated Approach to Software Engineering*, 4th ed., Springer, 2017.
4. S. Russell & P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2020.

Online resources :

1. NPTEL – Software Engineering – nptel.ac.in
2. MLflow Documentation – mlflow.org
3. Evidently AI – evidentlyai.com

4. Hugging Face – huggingface.co
5. SonarQube – sonarqube.org
6. GitHub Copilot – github.com/features/copilot
7. IEEE Software Engineering Standards – ieee.org

Recommended Free &Free-Tier AI/Database Tools :

1. **AI Assistants:** ChatGPT, Claude – scenario generation, SRS drafting, code review and documentation.
2. **Design & Modeling:** StarUML, Visual Paradigm, Figma, draw.io, Miro – UML, UI prototyping and workflow diagrams.
3. **Project & Docs:** Jira, Linear, Notion AI, Gamma.app – planning, estimation and documentation.
4. **Testing & Quality:** Testim, Selenium, SonarQube, Snyk, DeepCode – automated testing, static analysis and defect prediction.
5. **Low-Code / No-Code:** Mendix, OutSystems, Bubble, AppGyver – rapid application development.

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

II B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

(26CS0515) OPERATING SYSTEMS AND INTELLIGENT RESOURCE MANAGEMENT LAB

(Professional Core – IV)
(Common to CSE,IOT,CSD branches)

Prerequisites: Programming in C and Python; basic Linux familiarity. (Inferred – not specified in the source.)

Course Objectives :

The objectives of this course is to

1. *Develop practical understanding of the Linux environment, process management, memory management, file systems, and system resource utilization.*
2. *Apply scheduling, synchronization, deadlock handling, virtual memory, and disk management concepts.*
3. *Design concurrent and resource-management solutions using system calls and multithreading.*
4. *Integrate AI techniques for system monitoring and intelligent resource optimization.*
5. *Gain exposure to containerization and intelligent deployment environments.*

List of Experiments :

MODULE I: Linux Environment And Process Management

Topics: Linux installation and environment; file, directory, permission, and process commands; process creation using fork(), exec(), and wait(); process trees and parent/child memory comparison.

AI Integration: Explain Linux command functionality and outputs; generate practice exercises; explain process-creation sequence; generate process trees; debug process-creation errors.

Suggested Tools: Linux (Ubuntu), C (GCC), ChatGPT, Microsoft Copilot, GitHub Copilot

Experiments:

1. **Experiment 1: Linux Commands and Environment —** Install Ubuntu/Linux (physical, virtual, or cloud); execute file, directory, permission, and user/process management commands; create a project directory structure, modify permissions, and monitor running processes with CPU and memory utilization.
2. **Experiment 2: Process Creation using System Calls —** Write a C program using fork() to create a child process; display PID and PPID; use exec() and wait(); observe execution order, compare parent/child memory before and after exec(), and draw the process tree.

MODULE II: Cpu Scheduling, Synchronization & Multithreading

Topics: CPU scheduling — FCFS, SJF, Priority, Round Robin; Gantt charts and performance metrics; multithreading; producer–consumer, readers–writers, and dining-philosophers synchronization.

AI Integration: Generate process datasets and scenarios; verify scheduling calculations; generate Gantt charts; compare algorithm performance; detect race conditions and suggest synchronization improvements; generate synchronization traces.

Suggested Tools: C / Python, pthreads, semaphores / mutex, ChatGPT, GitHub Copilot

Experiments:

1. **Experiment 3: Process Scheduling – FCFS and SJF —** Read process information; implement FCFS and non-preemptive SJF scheduling; display Gantt charts, completion/waiting/turnaround times and averages; compare the two on average waiting time, turnaround time, and CPU utilization with a comparison graph.
2. **Experiment 4: Priority & Round Robin Scheduling —** Implement Priority and Round Robin scheduling; compare fairness and efficiency across scenarios.
3. **Experiment 5: Multithreading —** Create threads using pthreads/Python; execute concurrent tasks; compare sequential and parallel execution.
4. **Experiment 6: Producer–Consumer Synchronization —** Implement producer–consumer using semaphores/mutex with a shared buffer; detect and resolve race conditions.
5. **Experiment 7: Readers–Writers / Dining Philosophers —** Implement a classic synchronization problem while avoiding deadlock and starvation; analyse resource contention.

MODULE III : Memory Management And File Systems

Topics: Memory allocation — First Fit, Best Fit, Worst Fit; page replacement — FIFO, LRU, Optimal; Banker's Algorithm; file-system and directory operations.

AI Integration: Visualize allocation and compare utilization; simulate page replacement and compare efficiency; generate allocation scenarios and validate deadlock prevention; generate file structures and validate operations.

Suggested Tools: C / Python, Jupyter Notebook, Google Colab, ChatGPT, GitHub Copilot

Experiments:

1. **Experiment 8: Memory Allocation Techniques —** Implement First Fit, Best Fit, and Worst Fit; compare fragmentation.
2. **Experiment 9: Page Replacement Algorithms —** Implement FIFO, LRU, and Optimal; compute page faults and compare efficiency.
3. **Experiment 10: Banker's Algorithm —** Implement Banker's Algorithm; identify safe and unsafe states; validate deadlock prevention.
4. **Experiment 11: File System Operations —** Create, read, and write files; perform directory operations; simulate file allocation.

MODULE IV :Disk Management And System Analytics

Topics: Disk scheduling — FCFS, SSTF, SCAN, C-SCAN; system monitoring and log analysis.

AI Integration: Generate disk requests and compare performance; detect anomalies and interpret system behaviour.

Suggested Tools: C / Python, Jupyter Notebook, ChatGPT

Experiments:

1. **Experiment 12: Disk Scheduling Algorithms —** Implement FCFS, SSTF, SCAN, and C-SCAN; compare seek time.
2. **Experiment 13: System Monitoring & Log Analysis —** Monitor CPU and memory; analyse system logs; detect anomalies and interpret behaviour.

MODULE V: Intelligent Resource Management

Topics: Workload prediction using machine learning; containerization with Docker.

AI Integration: Generate visualizations and compare predictions; generate Dockerfiles and explain deployment.

Suggested Tools: Python, Scikit-learn, Google Colab, Docker, ChatGPT

Experiments:

1. **Experiment 14: Workload Prediction using ML —** Collect CPU-usage data; build a prediction model; predict workload and compare predictions.
2. **Experiment 15: Docker Containerization —** Install Docker; create and deploy a container; generate a Dockerfile and explain deployment.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Demonstrate Linux environment usage and process management using system calls.	L3 (Apply)
CO2	Apply CPU scheduling, synchronization, and multithreading concepts.	L3 (Apply)
CO3	Implement memory management, virtual memory, and file-system operations.	L3 (Apply)
CO4	Analyse disk scheduling and resource allocation mechanisms.	L4 (Analyze)
CO5	Develop intelligent OS applications using AI and container technologies.	L4 (Analyze)

CO – PO Articulation Matrix

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

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

Reference Books :

1. Silberschatz, A., Galvin, P. B., & Gagne, G., *Operating System Concepts*, 10th Ed., John Wiley & Sons, 2018.
2. Tanenbaum, A. S., & Bos, H., *Modern Operating Systems*, 4th Ed., Pearson, 2015.
3. Stallings, W., *Operating Systems: Internals and Design Principles*, 9th Ed., Pearson, 2018.
4. Das, S., *UNIX: Concepts and Applications*, 4th Ed., McGraw-Hill Education, 2017.
5. Love, R., *Linux System Programming*, 2nd Ed., O'Reilly Media, 2013.
6. Gregg, B., *Systems Performance: Enterprise and the Cloud*, 2nd Ed., Addison-Wesley Professional, 2020.

Online resources :

1. Poulton, N., *Docker Deep Dive*, Updated Ed., Nigel Poulton, 2023.
2. Géron, A., *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Ed., O'Reilly Media, 2022.
3. Linux Documentation Project – tldp.org
4. https://www.youtube.com/@JoyofLearningCSE/featured
5. Docker Documentation – docs.docker.com
6. Scikit-learn Documentation – scikit-learn.org
7. Jupyter Documentation – docs.jupyter.org
8. NPTEL – Operating Systems – nptel.ac.in
9. YouTube – Joy of Learning CSE – youtube.com/@JoyofLearningCSE

Recommended Free & Open-Source OS / AI Tools :

1. **Operating System:** Linux (Ubuntu), VirtualBox / VMware – environment and process experimentation.
2. **Programming:** C (GCC), Python, pthreads – system programming and concurrency.
3. **Notebooks & ML:** Jupyter Notebook, Google Colab, Scikit-learn – simulation and workload prediction.
4. **Containers:** Docker – containerization and deployment.
5. **AI Assistants:** ChatGPT, GitHub Copilot, Microsoft Copilot – code generation, debugging and explanation.

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

II B.Tech – II Sem.

L	T	Pr	P	C
0	0	0	3	1.5

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

Prerequisites: Data Structures.

Course Objectives :

The objectives of this course is to

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

List of Experiments :

MODULE I: Database Modelling And Sql Foundations

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

Suggested Tools: draw.io; ChatGPT

Experiments:

1. **Experiment 1: ER Modelling —** Design an ER diagram for a real-world application (e.g., library, hospital, or e-commerce system) and convert it into a relational schema using ER-to-relational mapping rules.
2. **Experiment 2: Database & DDL Operations —** Create a database and perform DDL operations (CREATE, ALTER, DROP) with integrity constraints such as Primary Key, Foreign Key, CHECK, UNIQUE, and NOT NULL.
3. **Experiment 3: DML & DQL Operations —** Perform DML and DQL commands (INSERT, UPDATE, DELETE, SELECT) with filtering, sorting, grouping, and aggregate functions on the created schema.

MODULE II: Sql Programming And Query Optimization

AI Integration: Use GitHub Copilot and ChatGPT to generate SQL queries and compare execution plans.

Suggested Tools: GitHub Copilot; ChatGPT

Experiments:

1. **Experiment 4: Joins & Subqueries —** Develop SQL queries using joins, nested queries, subqueries, set operations, aggregate functions, GROUP BY, and HAVING clauses.
2. **Experiment 5: Views, Indexes & Optimization —** Create and analyze views and indexes, and perform query optimization comparing indexed and non-indexed access methods.
3. **Experiment 6: Relational Algebra to SQL —** Implement relational algebra operations and convert relational algebra expressions into equivalent SQL statements.

MODULE III : Database Design And Normalization

AI Integration: Use ChatGPT and Jupyter Notebook to assist normalization and validate using Armstrong's axioms.

Suggested Tools: ChatGPT; Jupyter Notebook

Experiments:

1. **Experiment 7: Normalization (1NF–BCNF) —** Identify functional dependencies and perform normalization of relations from 1NF through BCNF.
2. **Experiment 8: Closures & Decomposition —** Implement attribute closure computation, minimal cover generation, lossless decomposition, and dependency preservation.
3. **Experiment 9: Schema Comparison —** Compare normalized and denormalized schema designs for analytical workloads and document the trade-offs.

MODULE IV : Transaction Management And Recovery

AI Integration: Use Python simulations and ChatGPT to generate transaction schedules and precedence graph analysis.

Suggested Tools: Python; ChatGPT

Experiments:

1. **Experiment 10: Transaction Control —** Implement transaction control operations using COMMIT, ROLLBACK, and SAVEPOINT.
2. **Experiment 11: Concurrency Control —** Simulate concurrency control mechanisms including locking protocols, two-phase locking, deadlock detection, and serializability analysis.
3. **Experiment 12: Logging & Recovery —** Implement database recovery mechanisms including logging, checkpointing, undo-redo recovery, and Write Ahead Logging.

MODULE V: Intelligent Data Management For Ai Applications

AI Integration: Use ChatGPT, LangChain, MongoDB Atlas, FAISS, Chroma, and Neo4j for intelligent database experimentation.

Suggested Tools: ChatGPT; LangChain; MongoDB Atlas; FAISS; Chroma; Neo4j

Experiments:

1. **Experiment 13: NoSQL with MongoDB —** Perform CRUD operations and aggregation queries using a NoSQL database such as MongoDB.
2. **Experiment 14: Graph Databases with Neo4j —** Develop a graph database application using Neo4j and implement Cypher queries.
3. **Experiment 15: Intelligent Applications —** Build intelligent database applications using Text-to-SQL, RAG-over-SQL, vector databases (FAISS/Chroma), similarity search, and AI-assisted data analytics.

Suggested Mini Projects:

1. An AI-assisted Text-to-SQL query interface for a chosen database schema.
2. A hybrid application combining a relational database (SQL) with a vector database (FAISS/Chroma) for semantic search.

Course Outcomes (COs) :

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

CO	Course Outcome Statement	Bloom's Level
CO1	Design ER models and convert them into relational schemas for real-world applications.	L3 (Apply)
CO2	Construct and execute SQL queries for data definition, manipulation, retrieval, and optimization.	L3 (Apply)
CO3	Apply normalization techniques and schema decomposition methods to design efficient databases.	L3 (Apply)
CO4	Implement transaction management, concurrency control, and recovery mechanisms in database environments.	L4 (Analyze)
CO5	Develop intelligent database applications using NoSQL systems, vector databases, and AI-assisted querying techniques.	L4 (Analyze)

CO – PO Articulation Matrix :

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

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

Textbooks :

1. Silberschatz, A., Korth, H. F., & Sudarshan, S., *Database System Concepts*, 7th Ed., McGraw-Hill Education, 2020.
2. Elmasri, R., & Navathe, S. B., *Fundamentals of Database Systems*, 7th Ed., Pearson, 2016.

Reference Books :

1. Ramakrishnan, R., & Gehrke, J., *Database Management Systems*, 3rd Ed., McGraw-Hill Education, 2003.
2. Kleppmann, M., *Designing Data-Intensive Applications*, O'Reilly Media, 2017.

Online resources :

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

Recommended Free & Open-Source AI/Computational Tools

1. **AI Assistants:** ChatGPT; GitHub Copilot – query generation, debugging, and schema/normalization validation.
2. **Modelling Tools:** draw.io – ER diagram design.
3. **Databases & Engines:** MongoDB (Atlas free tier); Neo4j (CommUnity/AuraDB Free) – NoSQL and graph database experimentation.
4. **Vector Search Libraries:** FAISS; Chroma – similarity search and vector database applications.
5. **Frameworks:** LangChain – building Text-to-SQL and RAG-over-SQL pipelines.
6. **Notebooks & Simulation:** Python & Jupyter Notebook – transaction simulations and normalization/closure computations.
7. TensorFlow / Keras (free, open-source) – For optional AI demonstrations in regression and classification tasks.

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

L	T	Pr	P	C
1	0	0	2	2

II B.Tech – II Sem

(26CS0571) DATA ANALYTICS USING PYTHON
(Skill Enhancement Course – II)
(Common to CSE & Allied branches)

Prerequisites: Basic knowledge of Python programming and programming fundamentals. (Inferred – not specified in the source.)

Course Objectives :

The objectives of this course is to

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

List of Experiments :

MODULE I: Data Handling Using Python

Topics: Data Analytics Workflow; Jupyter Notebook Environment; Data Sources and File Formats; Importing Data; CSV, Excel and JSON Processing; NumPy Arrays; Data Manipulation; Vectorized Operations; Data Export.

AI Integration: Dataset understanding, automatic profiling and analytical assistance.

Suggested Tools: Python, Jupyter Notebook, Pandas, NumPy

Experiments:

1. **Experiment 1: Dataset Import and Exploration —** Import datasets from CSV, Excel and JSON formats. Examine dimensions, data types, missing values and generate summary statistics.
2. **Experiment 2: Numerical Data Processing using NumPy —** Perform array creation, indexing, slicing, aggregation, mathematical operations and vectorized computations.
3. **Experiment 3: DataFrame Operations using Pandas —** Create DataFrames and perform selection, filtering, indexing, sorting and aggregation.

MODULE II: Data Processing And Transformation Using Pandas

Topics: Series and DataFrames; Data Selection; Indexing and Filtering; Missing Data Handling; Data Cleaning; Aggregation; Grouping; Merging; Sorting; Feature Construction (without ML).

AI Integration: AI-assisted cleaning, transformation recommendations and optimization.

Suggested Tools: Pandas, Jupyter Notebook

Experiments:

1. **Experiment 4: Data Cleaning and Missing Value Handling —** Identify duplicate records, missing values and outliers. Apply cleaning and preprocessing techniques.
2. **Experiment 5: Data Transformation and Feature Construction —** Perform grouping, merging, aggregation, reshaping and derived column generation.
3. **Experiment 6: Descriptive Statistics and Data Analysis —** Generate statistical summaries and perform exploratory analysis including Mean, Median, Mode, Variance and Distribution Analysis.

MODULE III: Exploratory Data Analysis And Visualization

Topics: Descriptive Statistics; Data Exploration; Correlation Analysis; Data Visualization Principles; Line Charts; Bar Charts; Scatter Plots; Histograms; Boxplots; Heatmaps; Interactive Visualization.

AI Integration: Visualization recommendation, dashboard generation and insight generation.

Suggested Tools: Matplotlib, Seaborn, Plotly

Experiments:

1. **Experiment 7: Data Visualization using Matplotlib —** Generate visual reports using line charts, bar charts, histograms and pie charts.
2. **Experiment 8: Advanced Visualization using Seaborn —** Develop scatter plots, boxplots, heatmaps and distribution analysis.
3. **Experiment 9: Interactive Visualization and Dashboard Design —** Build interactive visualizations and analytical dashboards.
4. **Experiment 10: Correlation and Comparative Data Analysis —** Perform comparative analysis and identify relationships among variables.

MODULE IV: Data Analytics Using Real-World Datasets

Topics: Data Collection Methods; Public Dataset Usage; JSON Processing; API-based Data Collection; Time-Series Data Handling; Data Transformation; Report Generation; Dashboard Concepts.

AI Integration: Automated analytics, report generation and insight extraction.

Suggested Tools: Pandas, Requests, Plotly

Experiments:

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

3. **Experiment 13: Analytical Report Generation —** Generate analytical summaries and reports from datasets.

MODULE V: Applied Data Analytics Project

Topics: Analytics Workflow; Data Preparation; Data Visualization; Insight Generation; Dashboard Development; Documentation; Project Presentation.

AI Integration: Data profiling, visualization recommendation, reporting and documentation.

Suggested Tools: Python, Jupyter Notebook, Plotly

Experiments:

1. **Experiment 14: Dashboard Development using Real-world Datasets —** Create complete dashboards for decision support using Education, Healthcare, Retail or Weather datasets.
2. **Experiment 15: Applied Data Analytics Mini Project —** Develop and present a complete analytics solution including data collection, data cleaning, analysis, visualization, dashboard and report generation.

Suggested Mini Projects:

1. Student Performance Analytics
2. Retail Sales Analytics
3. Healthcare Analytics
4. Weather Analytics
5. Financial Data Analytics
6. Social Media Analytics
7. Sports Analytics
8. Survey Data Analytics

Course Outcomes (COs) :

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

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

CO – PO Articulation Matrix

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

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

Textbooks :

1. Wes McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy and Jupyter*, O'Reilly Media.
2. Jake VanderPlas, *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media.
3. Joel Grus, *Data Science from Scratch: First Principles with Python*, O'Reilly Media.

Reference books :

1. Cathy O'Neil and Rachel Schutt, *Doing Data Science*, O'Reilly Media.
2. Trevor Hastie, Robert Tibshirani and Gareth James, *An Introduction to Statistical Learning*, Springer.
3. Allen B. Downey, *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media.
4. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, O'Reilly Media (selected chapters on data preparation and analysis).
5. Ethem Alpaydin, *Introduction to Machine Learning*, MIT Press (reference for analytics foundation).

Online learning resources :

1. Python Documentation – docs.python.org
2. Jupyter Documentation – docs.jupyter.org
3. NumPy Documentation – numpy.org/doc
4. Pandas Documentation – pandas.pydata.org/docs
5. Matplotlib Documentation – matplotlib.org/stable
6. Seaborn Documentation – seaborn.pydata.org
7. Plotly Documentation – plotly.com/python

Recommended Free & Open-Source AI/Computational Tools :

1. **Core Libraries:** Python, Pandas, NumPy – data handling, manipulation and numerical computing.
2. **Visualization:** Matplotlib, Seaborn, Plotly – static and interactive charts and dashboards.

3. **Environment:** Jupyter Notebook – interactive analysis, experimentation and reporting.
4. **Data Access:** Requests – API-based data collection and JSON processing.
5. **AI Assistants:** ChatGPT, Claude, GitHub Copilot – code generation, automatic data profiling and analytical assistance.

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

II B.Tech – II Sem.

L	T	Pr	P	C
2	0	0	0	0

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

Prerequisites: Communicative English

Course Objectives :

The objectives of this course is to

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

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

Unit-II :	**Technical Report Writing**
Content	Structure of a Formal Report: Title page, Abstract, Table of Contents, Introduction, Methodology, Results, Discussion, Conclusion, References, Appendices.

	Abstract Writing: Informative vs. Descriptive abstracts; Literature Review: Searching, evaluating, synthesising sources. Data Presentation: Tables, Figures, Charts; Citation and Referencing: IEEE, APA, and MLA styles; avoiding plagiarism.
AI Integration	AI Tool: Elicit.org for AI-powered literature search. AI Tool: Zotero for citation management and bibliography generation. AI Tool: ChatGPT for first-draft abstract generation. Plagiarism check using Turnitin or Copyleaks.

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

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

Unit-V :	**Patent Drafting and IP Strategy**
Content	Structure of a Patent Application: Title, Abstract, Background, Summary, Detailed Description, Claims, Drawings; Types of Claims: Independent and Dependent claims. Filing a Provisional Patent Application (PPA) in India; IP Strategy for Startups and Engineers: trade secrets vs. patents, licensing, technology transfer.

	Case Studies: Famous patent disputes (Apple vs. Samsung, Monsanto seed patents, pharmaceutical patents).
AI Integration	AI Tool: ChatGPT to draft patent claims for a mechanical invention and critique for clarity. AI Tool: IP India Online Portal navigation. Capstone: write a complete patent disclosure document.

Bloom's Taxonomy Coverage – Summary

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

Course Outcomes (COs) :

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

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

CO – PO / PSO Articulation Matrix (NBA)

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

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	1	1	1	1	1	1	1	**2**	1	**3**	**2**	**2**	**2**	**2**
CO2	**2**	**2**	**2**	**2**	**2**	1	1	**2**	**2**	**3**	**2**	**3**	**2**	**2**
CO3	**2**	**3**	**2**	**2**	**3**	1	1	**2**	**2**	**3**	**2**	**3**	**2**	**2**

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

Textbooks:

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

Reference Books :

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