JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY: KAKINADA

(Established by Andhra Pradesh Act No.30 of 2008)

Kakinada – 533 003, Andhra Pradesh (India)

Vision and Mission of the University

Vision

The University is primarily promoting quality of education in the areas of Science, Technology, Engineering, Mathematics (STEM) and also Management disciplines, as academic pillars of education, to excel in teaching, learning, research, consultancy, placements through innovative practices with a global perspective and entrepreneurship skills

Mission

The University is moving forward with a mission of promoting excellence, access, inclusion, and relevance in quality in Science, Technology, Engineering, Mathematics, and Management disciplines, viz.,

- M1: Design an industry-relevant curriculum from time to time with a Global perspective.
- M2: Promote quality education by embracing ICT delivery mechanisms with continuous pedagogy through e-learning mechanisms.
- M3: Spread across for industry collaborations with a focus on pre-training and placements for technology transfer to society.
- M4: Establishing centres of excellence to promote research and innovations in multidisciplinary areas to bring in patent culture and consultancy practices.
- M5: International Collaborations for student outreach, facilitating international students to study in JNTUK to infuse cross-cultural learning practices.
- M6: Academic Excellence, Research & Innovation, People Technology & Creativity, Social & Environmental Responsibility, Human & Intellectual Knowledge.

Vision and Mission of the Institute

Vision and Mission of the Department

B.Tech. –

CSE (Internet of Things)

Program Course Structure

&

Syllabus

Program Educational Objectives (PEOs)

Program Educational Objectives are broad statements that describe the career and professional accomplishments the program prepares graduates to achieve.

- **PEO1:**Apply strong foundations in computing and IoT engineering to design connected systems that solve real-world problems across diverse domains.
- **PEO2:**Excel in professional careers spanning software development, IoT product engineering, data analytics, and cybersecurity, or pursue higher studies and research.
- **PEO3:**Adapt to emerging technologies through continuous learning and deliver innovative, sustainable solutions to interdisciplinary challenges
- **PEO4:**Practice their profession with integrity, effective communication, and teamwork, showing responsibility toward societal, environmental, and ethical concerns.

Mapping of PEOs with Mission Statements

	PEO1	PEO2	PEO3	PEO4
M1	✓		✓	
M2	✓		✓	
M3	✓	✓	✓	✓
M4	✓		✓	✓
M5		✓	✓	✓
M6	✓	✓	✓	✓

Legend: ✓ = Strong Alignment (Green Shading) | No Mark = Weak/No Alignment (Orange Shading)

Program Outcomes (POs)

POs are statements that describe what students are expected to know and be able to do upon completion of the program (As defined by NBA).

Program Outcomes (POs)

- **PO1: Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization to develop solutions to complex engineering problems.
- **PO2: Problem Analysis**: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.
- **PO3: Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.
- **PO4: Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.
- **PO5: Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.
- **PO6: The Engineer and The World:**Analyze and evaluate societal and environmental aspects while solving complex engineering problems for their impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.
- **PO7: Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.
- **PO8: Individual and Collaborative Teamwork:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
- **PO9: Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
- **PO10: Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
- **PO11: Life-Long Learning:** Recognize the need for, and have the preparation and ability for
  i) independent and life-long learning
  ii) adaptability to new and emerging technologies and
  iii) critical thinking in the broadest context of technological change.

Program Specific Outcomes (PSOs)

- **PSO1:**Design and deploy end-to-end IoT solutions by integrating sensors, embedded systems, communication protocols, and cloud/edge platforms.
- **PSO2:**Apply data analytics, AI, and machine learning techniques to derive actionable insights from IoT-generated data.
- **PSO3:**Develop secure, scalable, and reliable IoT systems by addressing cybersecurity, privacy, and ethical considerations.

B.Tech. – – I Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	HSM		English for Technical Communication	3	0	0	3
2	BSC		Applied Chemistry for Emerging Technologies	3	0	0	3
3	BSC		Linear Algebra and Calculus	3	0	0	3
4	ESC		Computational Thinking and Problem Solving using C	3	0	0	3
5	ESC		Fundamentals of Digital Logic	2	0	0	2
6	HSM		English for Technical Communication and MOODLE Lab	0	0	3	1.5
7	BSC		Applied Chemistry for Emerging Technologies Lab	0	0	3	1.5
8	ESC		Computational Thinking and Problem Solving using C Lab	0	0	3	1.5
9	ESC		IT Workshop	0	0	3	1.5
10	MNC		Sports and Yoga or NSS/NCC	0	0	2	0
		Total		**14**	**0**	**14**	**20**

B.Tech – IYear II Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	BSC		Physics for Advanced Computing	3	0	0	3
2	BSC		Differential Equations and Vector Calculus	3	0	0	3
3	ESC		Data Structures using C	3	0	0	3
4	ESC		Python Programming	2	0	0	2
5	ESC		Sustainability and Green Engineering	2	0	0	2
6	ESC		Data Structures using C Lab	0	0	3	1.5
7	ESC		Design Thinking and Innovation	0	1	3	2.5
8	ESC		AI Tools and Applications	0	1	3	2.5
9	BSC		Physics for Advanced Computing Lab	0	0	3	1.5
10	ESC		Python Programming Lab	0	0	2	1.0
11	MNC		Indian Knowledge Systems	2	0	0	0
		Total		**15**	**2**	**14**	**22**

B.Tech. – – II Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	BSC		Discrete Mathematics and Graph Theory	3	0	0	3
2	HSM		Managerial Economics and Financial Analysis	3	0	0	3
3	PCC		Fundamentals of IoT	3	0	0	3
4	PCC		Operating Systems	3	0	0	3
5	PCC		Software Engineering	3	0	0	3
6	PCC		IoT Protocols and Communication	3	0	0	3
7	PCC		Fundamentals of IoT Lab	0	0	3	1.5
8	PCC		Operating Systems Lab	0	0	3	1.5
9	MNC		Universal Human Values II	2	0	0	0
Total				**20**	**0**	**06**	**21**

B.Tech. – -II Year II Semester

S.No	Course category	Course Code	Title	L	T	P	Credits
1	BSC		Probability and Statistics	3	0	0	3
2	PCC		Computer Organization	3	0	0	3
3	PCC		Web Technologies and Design	3	0	0	3
4	PCC		Computer Networks	3	0	0	3
5	PCC		Sensors and Actuators for IoT	3	0	0	3
6	PEC		**Professional Elective -1** 1. Smart City Architecture and Planning 2. IoT for Energy Management 3. IoT for Patient Monitoring Systems 4. Precision Agriculture IoT	3	0	0	3
7	PCC		Computer Networks Lab	0	0	3	1.5
8	PCC		Sensors and Actuators Lab	0	0	3	1.5
9	MNC		Indian Constitution	2	0	0	0
Total				**20**	**0**	**06**	**21**
Mandatory Industry Internship / CSP / Mini Project of 06 weeks duration during summer vacation							

B.Tech. – - III Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	PCC		Embedded Systems and Microcontrollers	3	0	0	3
2	PCC		Data Base Management Systems	3	0	0	3
3	PCC		IoT Security and Privacy	3	0	0	3
4	PEC		**Professional Elective-2** 1. Autonomous Vehicles IoT 2. Edge Computing for Real-time Robotics 3. Environmental Monitoring with IoT 4. Industry 4.0 and Smart Manufacturing	3	0	0	3
5	PEC		**Professional Elective-3** 1 Supply Chain IoT 2 Industrial Control Systems and SCADA 3 Climate and Weather Prediction Systems 4 SWAYAM / NPTEL 12Week MOOC	3	0	0	3
6	PCC		Embedded Systems and IoT Security Lab	0	0	3	1.5
7	PCC		Data Base Management SystemsLab	0	0	3	1.5
8	OEC		Open Elective-1	1	0	0	1
9	Internship / Project work		Evaluation of Industry Internship/ CSP / Mini Project*	0	0	0	2
10	MNC		Environmental Studies	2	0	0	0
		Total		**18**	**0**	**6**	**21**

* performed during IV to V Semester break

Courses offered under Open Elective-1 (for other department students):

1. Data Structures
2. Principles of Software Engineering

B.Tech. – – III Year II Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	PCC		IoT Architecture and Design	3	0	0	3
2	PCC		Wireless Sensor Networks	3	0	0	3
3	PCC		Big Data Analytics for IoT	3	0	0	3
4	PEC		**Professional Elective-4** 1 Industrial IoT Networking 2 Arduino-based IoT 3 Cyber Security 4 Cryptography for IoT	3	0	0	3
5	PEC		**Professional Elective-5** 1 Industrial IoT Architecture 2 Data Visualization 3 Blockchain for IoT Security 4 IoT Quality of Service (QoS)	3	0	0	3
6	OEC		Open Elective-2 / SWAYAM / NPTEL 8 week MOOC	2	0	0	2
7	PCC		IoT Design Lab	0	0	3	1.5
8	PCC		Wireless Sensor NetworksLab	0	0	3	1.5
9	OEC		Open Elective-3	0/1	0	2/0	1
10	MNC		Professional Practice, Law and Ethics	2	0	0	0
			Total	**20**	**0**	**6**	**21**
Industry Internship / Mini Project of 06 weeks duration during summer vacation							

Courses offered under Open Elective-2 (for other department students):

1. Principles of DataBase Management Systems
2. Artificial Intelligence

Courses offered under Open Elective-3 (for other department students):

1. Object Oriented Programming Through JAVA (L-T-P- 0-0-2)
2. Human Computer Interaction (L-T-P- 1-0-0)

B.Tech. – -IV Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	PCC		Quantum Computing	3	0	0	3
2	PCC		Generative AI	3	0	0	3
3	PEC		**Professional Elective-6** 1. Cloud Computing for IoT 2. Smart Home Automation Systems 3. Urban Computing 4. Predictive Maintenance and Anomaly Detection	3	0	0	3
4	PEC		**Professional Elective-7** 1 Drone Technology and UAV Systems 2 Robotics and Autonomous Systems 3 Raspberry Pi-based IoT 4 IoT Malware Analysis	3	0	0	3
5	PEC		**Professional Elective-8** 1 Healthcare IoT and Telemedicine Systems 2 Wearable Technology Design 3 Biomedical Signal Processing 4 Smart Water Management Systems	3	0	0	3
6	PEC		**Professional Elective-9**	0	0	2	1
7	OEC		Open Elective-4	2	0	0	2
8	OEC		Open Elective-5	0	0	2	1
9	Internship / Project work		Evaluation of Industry Internship / Mini Project	0	0	0	2
			Total	**17**	**0**	**4**	**21**

*Includes evaluation of mini project

Courses offered under Professional Elective-9

1. User Interface Design Using Flutter
2. Courses offered by SWAYAM Plus

Courses offered under Open Elective-4 (for other department students):

1. Social Media Analytics
2. Cyber Security

Courses offered under Open Elective-5 (for other department students):

1. Data Visualization
2. Internet of Things

B.Tech. – – IV Year II Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	OEC		Open Elective 6 / SWAYAM/NPTEL 12 week course	3	0	0	3
2	OEC		Open Elective 7 / SWAYAM/NPTEL 8 week course	2	0	0	2
3	Project		Project work / Industry Internship				12
			Total				17

Courses offered under Open Elective-6 (for other department students):
1. Quantum computing
2. Data Science
3. SWAYAM / NPTEL MOOC (BoS need to suggest the courses)

Courses offered under Open Elective-7 (for other department students):
1. Cloud Computing
2. Generative AI
3. SWAYAM / NPTEL MOOC (BoS need to suggest the courses)

Suggested Courses for **Minor(s)** – 18 Credits
1. (II-I) (3-0-3)
2. (II-II) (3-0-0)
3. (III-I) (3-0-3)
4. (III-II) (3-0-0)
5. (IV-I) (3-0-0)

Suggested Courses for **Honor's** – 18 Credits
1. (III-I) (3-0-0)
2. (III-I) (3-0-0)
3. (III-II) (3-0-0)
4. (III-II) (3-0-0)
5. (IV-I) (3-0-0)
6. (IV-I) (3-0-0)

Credits Distribution – Based on Category (Credits per year - for 1, 2, 3 years are - 42 ONLY)

	I year		II year		III year		IV year			
Category/ Semester	I	II	III	IV	V	VI	VII	VIII	Credits	%
BSC	7.5	7.5	3	3	0	0	0	0	21	12.8
HSM	4.5	0	3	0	0	0	0	0	7.5	4.6
ESC	8	14.5	0	0	0	0	0	0	22.5	13.7
PCC	0	0	15	15	12	12	6	0	60	36.6
PEC	0	0	0	3	6	6	10	0	25	15.2
OEC	0	0	0	0	1	3	3	5	12	7.3
Internship & Project work	0	0	0	0	2	0	2	12	16	9.8
MNC	0	0	0	0	0	0	0	0	0	0
Total	20	22	21	21	21	21	21	17	**164**	100

Category	3 Credit	2.5 credit	2 credit	1.5 credit	1 credit	Total
PCC	16	0	0	0	0	48
PCC Lab	0	0	0	8	0	12
PEC	8	0	0	0	1	25
OEC	1	0	3	0	3	12

COURSE CODE	ENGLISH FOR TECHNICAL COMMUNICATION	Category HSM	L-T-P-C 3-0-0-3

Pre-requisites: NIL

Course Objectives:

- Listen the authentic texts to comprehend and respond appropriately as both audience and conversant;
- Communicate in academic and social (formal) contexts with reasonable fluency and accuracy (neutral accent, social and grammatical appropriateness);
- Comprehend the texts of various types for academic, technical and career purposes;
- Write for academic and professional purposes;
- Communicate across cultures that requires a blend of cognitive, interpersonal, and linguistic skills

Course Outcomes:

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Comprehend specific information, context, and central ideas from social, academic, or workplace conversations and audio-visual material.	K2
CO2	Apply appropriate social norms to bridge the gaps and collaborate effectively across diverse cultural backgrounds	K3
CO3	Examine discourse features and speech strategies to participate effectively in group discussions, informal exchanges, and short presentations on technical and general topics.	K4
CO4	Evaluate the content, limitations and ethical aspects of AI tools in communication and language learning, and apply them to create concise, accurate essays/summaries.	K5
CO5	Construct well-organized paragraphs, essays, critical reviews, and other professional documents suited to academic and workplace contexts.	K6

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	L	M	–	–	L	M	–	–	L	H	L			
CO2	L	L	L	–	L	H	M	M	H	H	M			
CO3	L	M	L	L	L	M	L	L	H	H	M			
CO4	M	H	M	M	H	L	L	H	M	H	H			
CO5	L	M	L	L	L	M	L	L	H	H	H			

UNIT	CONTENTS	Contact Hours
UNIT - 1	Effective Communication: Exploring Channels and Methods Vocabulary: Roots, Prefixes, and Suffixes Homonyms Grammar: Articles, Prepositions, Tenses. Reading: Definition and Importance, Skills and Sub-Skills of Reading Writing: Cohesion and Coherence, Paragraph Writing. Listening: Listening for main ideas and supporting details, Note-taking Speaking: Introducing self and others, Describing people and objects, JAM.	
UNIT - 2	Eco-Consciousness: Negotiating for a Sustainable Future Vocabulary: Contractions, Phrasal Verbs, Idioms & Phrases Grammar: Subject-Verb Agreement, Conditional Sentences Reading: Reading for Inference, Point of View,	

UNIT	CONTENTS	Contact Hours
	Tone and Perspective. Writing: Note making, Summarizing, Paraphrasing. Listening: Listening to various texts and comprehension Speaking: Role-plays, Extempore.	
UNIT - 3	Disruptive Technologies: Ethical Implications and Future Prospects Vocabulary: One-Word Substitutes, Commonly Confused Words Grammar: Degrees of Comparison, Active and Passive Voice, Question Tags. Reading: Reading the extracts/articles/ essays of the News Papers, Journals and literary texts. Writing: Paraphrasing, Letter Writing Listening: Listening to stories, news, documentaries and discussions. Speaking: Debate, Group Discussions	
UNIT - 4	Entrepreneurship and Start-Up Culture: Emerging Trends Vocabulary: Business/ Profession/ Career/ Job-related Vocabulary. Grammar: Sentence Synthesis, Simple, Compound and Complex Sentences. Reading: Reading various texts for global understanding Writing: Email Writing, Data Interpretation, Report Writing Listening: Listening to debates and discussions. Speaking: Power Point Presentations by Individuals and Teams.	
UNIT - 5	Ethical Leadership: Integrating Social Responsibility and Core Values Vocabulary: Advanced Vocabulary for GRE/TOEFL/ IELTS/CAT etc. Grammar: Sentence Synthesis, Para Jumbles. Reading: Reading texts for global comprehension. Writing: Book/Film Reviews, Social Media Etiquette Listening: Listening to the talks/speeches by the eminent personalities Speaking: Team Discussions and Presentations	

Learning Resources:

Textbook:

1. Chekuri, T., & Perumalla, R. K. (2026). NextGen English for engineers: A course book for technical English communication. Cengage.
2. Coelho, P. (2020). The alchemist. HarperCollins. (Non-Detailed Text)

Reference Books:

1. Technical Communication (2nd Ed.) by M. Raman & S. Sharma. Oxford University Press, 2012.
2. Effective Technical Communication by M.A. Rizvi, Tata McGraw-Hill, 2005.
3. English for Engineers and Technologists, Orient Blackswan 2010 (Combined edition, Vol.1 &2).
4. The Oxford Guide to Writing and Speaking by John Seely, Oxford University Press, 2004.

Web Resources:

1. eGyanKosh. (n.d.). National digital repository. https://egyankosh.ac.in/
2. Read Theory. (n.d). Adaptive reading comprehension practice. https://readtheory.org/

COURSE CODE	APPLIED CHEMISTRY FOR EMERGING TECHNOLOGIES	Category BSC	L-T-P-C 3-0-0-3

Pre-requisite: NIL

Course Objectives

- Understand the fundamentals of quantum chemistry, including wave-particle duality, quantum numbers, and molecular orbital theory relevant to computing materials.
- Comprehend the chemistry and electronic properties of semiconductor and superconductor materials and the chemical processes involved in semiconductor-device fabrication.
- Evaluate electrochemical principles governing batteries, supercapacitors, and fuel cells as energy-storage technologies for computing systems.
- Design sensors and biosensors and utilize them in computer applications.
- Explore the chemistry, synthesis methods, and properties of nanomaterials used in nanoelectronics, quantum computing, and sustainable electronics.

Course Outcomes (COs)

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Apply quantum chemistry principles to analyze molecular systems and explain molecular orbital theory for computing materials.	K3, K4
CO2	Analyze semiconductor materials, doping chemistry and fabrication processes used in modern electronic devices.	K3, K4
CO3	Evaluate energy-storage technologies including batteries, super capacitors and fuel cells for computing and data-centre applications.	K3, K4
CO4	Design sensors and utilize smart materials for modern devices.	K3, K4
CO5	Design advanced materials solutions using nanotechnology principles and AI-assisted tools for Nano electronics.	K5, K6

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	M	L	M	L	-	-	-	-	-	M			
CO2	H	H	M	M	M	-	-	-	-	-	M			
CO3	H	H	M	M	M	M	L	-	-	L	M			
CO4	M	H	H	M	M	M	L	-	-	L	M			
CO5	H	M	H	H	H	M	L	L	-	L	H			

UNIT	CONTENTS	Contact Hours
UNIT - 1	Quantum Chemistry 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	9

UNIT	CONTENTS	Contact Hours
	heteronuclear diatomic molecules; energy-level diagrams of O2, N2, CO and HF; π-molecular orbitals of butadiene and benzene; calculation of bond order.	
UNIT - 2	Chemistry of Semiconductors and Superconductors Band theory of semiconductors, Intrinsic and extrinsic semiconductors, Preparation of pure and doped semiconductor material, silicon dioxide (SiO2) formation and properties, Chemistry of compound semiconductors (GaAs, InP, SiC), Advanced semiconductor materials (Organic semiconductors, Perovskite semiconductors) Photolithography: Basic principle, applications Superconductors: Characteristics, Type-I and Type-II, Low temperature and high temperature superconductors, superconducting materials (NbTi, Nb3Sn), applications Ion-trap materials: Basic concept, applications	10
UNIT - 3	Triad of Electrochemical Power Galvanic and electrolytic cells, redox reactions, electrode potential, Nernst equation and problems; Primary batteries: Dry cell, Lithium battery, sodium-air batteries (construction, working and applications), zinc-air battery, secondary batteries: lead-acid, Ni-Cd, lithium-ion, sodium-ion batteries (construction, charging-discharging reactions and applications). Battery management systems (main functions, basic working, Battery protection) Fuel cells: PEMFC, DMFC (working, advantages, applications) Supercapacitors: EDLC, pseudo-capacitors, hybrid supercapacitors for energy harvesting, applications	9
UNIT - 4	Sensors, Biosensors and smart materials E-Waste and Recycling: Recovery of metals, Battery recycling, Safe disposal, Reuse of electronic materials, Reducing environmental contamination Sensors: Basic Working Principle, Types of Sensors (Gas sensors, Temperature sensors, Pressure sensors, pH sensors, Humidity sensors, Chemical sensors) Biosensors: Components of a Biosensor, Applications Smart materials: Types and applications (Shape-memory alloys, Piezoelectric materials, Electrochromic materials, Self-healing polymers)	9
UNIT - 5	Chemistry of nanomaterials Introduction to Nanochemistry Carbon nanomaterials: Fullerenes, carbon nanotubes (CNTs), graphene; synthesis method [chemical vapor deposition (CVD)], Applications. Quantum dots: Synthesis, properties, applications in displays and real time applications. Metallic nanoparticles: Preparation (sol-gel process) and applications of gold, silver, and copper nanoparticles. Applications in nanoelectronics, interconnects, sensors (optical and biosensors), memory devices. Spintronics materials: Magnetic nanoparticles, topological insulators.	9
	Total	**46**

Learning Resources

Textbooks

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

Reference Books

1. Linden, D., & Reddy, T. B. (Eds.). (2001). Handbook of batteries (3rd ed.). McGraw-Hill.
2. Rao, C. N. R., Müller, A., & Cheetham, A. K. (Eds.). (2007). The chemistry of nanomaterials: Synthesis, properties and applications (Vols. 1–2). Wiley-VCH.
3. Sharma, B. K., & Gaur, H. (2019). Engineering chemistry. Krishna Prakashan Media.

Web Resources

- NPTEL. (n.d.). Engineering chemistry courses. https://nptel.ac.in
- IBM Quantum. (n.d.). IBM Qiskit learning. https://learning.quantum.ibm.com
- Google. (n.d.). Google Colab. https://colab.research.google.com
- OpenCV. (n.d.). OpenCV Python documentation. https://docs.opencv.org
- Materials Project. (n.d.). Open nanomaterials database. https://materialsproject.org

COURSE CODE	LINEAR ALGEBRA AND CALCULUS	Category BSC	L-T-P-C 3-0-0-3

Pre-requisite: Mathematics of 10+2 / Intermediate level (algebra, matrices and basic calculus)

Course Objectives:

- The main objective of this course is to provide the learner with variety of linear algebra techniques to solve linear systems, orthogonal representation of the quadratic surfaces and apply its knowledge in optimization problems.
- It aims further to calculate areas, surfaces and volumes in various coordinate systems.

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Apply matrix algebra and numerical techniques to determine the rank, inverse, eigenvalues and eigenvectors of matrices, solve homogeneous and non-homogeneous systems of linear equations using direct and iterative methods, and analyze their applications in engineering and scientific problems	K2, K3, K5
CO2	Analyze and solve engineering problems by applying matrix diagonalization, Singular Value Decomposition (SVD), the Cayley-Hamilton Theorem for matrix inverses and powers, and orthogonal transformations to classify and reduce quadratic forms to canonical forms	K2, K3, K5
CO3	Find the mean value of a given function and use it to estimate the bounds of a given function	K2, K3, K5
CO4	Compute change in the dependent variable with respect to many independent variables and Jacobians. Apply the knowledge in optimizing functions of several variables	K2, K3, K5
CO5	Compute double and triple integrals of functions of several variables and must be able to change the coordinate systems from Cartesian to polar, spherical and cylindrical coordinates	K2, K3, K5

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	H	M											
CO2	H	H	M											
CO3	H	H	M											
CO4	H	H	M											
CO5	H	H	M											

UNIT	CONTENTS	Contact Hours
UNIT - 1	Matrices, Eigenvalues, Eigenvectors: Rank of a matrix by echelon form, normal form. Inverse of Non-singular matrices by Gauss-Jordan method, System of linear equations: Solving system of Homogeneous and Non-Homogeneous equations, Gauss elimination method, Jacobi and Gauss Seidel Iteration Methods. Eigenvalues, Eigenvectors and their properties.	

UNIT	CONTENTS	Contact Hours
UNIT - 2	Diagonalization of a Matrix: Diagonalization of a matrix, Cayley-Hamilton Theorem (without proof), Finding inverse and power of a matrix by Cayley-Hamilton Theorem, Quadratic forms and Nature of the Quadratic Forms, Reduction of Quadratic form to canonical forms by Orthogonal Transformation. Singular valued Decomposition.	
UNIT - 3	Calculus: Mean Value Theorems: Rolle's Theorem, Lagrange's mean value theorem with their geometrical interpretation, Cauchy's mean value theorem, Taylor's and Maclaurin theorems with remainders (without proof), Problems and applications on the above theorems. Taylor's and Maclaurin series.	
UNIT - 4	Partial Differentiation and Applications (Multi Variable Calculus): Functions of several variables: Continuity and Differentiability, Partial derivatives, Total derivatives, Chain rule, Euler's Theorem, Taylor's and Maclaurin's series expansion of functions of two variables. Jacobians, Functional dependence, Maxima and Minima of functions of two variables without constraints and with Constraints-Method of Lagrange multipliers.	
UNIT - 5	Multiple Integrals: Double integrals, Change of order of integration, Triple integrals, Change of variables from Cartesian to polar, Cylindrical and Spherical coordinates. Finding areas (by double integrals) and volumes (by double integrals and triple integrals).	

Learning Resources:

Text Books:

1. Grewal, B.S. (2024). Higher Engineering Mathematics, 45th Edition. Khanna Publishers.
2. Kreyszig, E. (2025). Advanced Engineering Mathematics, 11th Edition. John Wiley & Sons.

Reference Books:

1. Ramana, B.V. (2017). Higher Engineering Mathematics. McGraw Hill Education.
2. James, G. (2018). Advanced Modern Engineering Mathematics. 5th Edition Pearson publishers.
3. Dass, H.K. (2022). Advanced Engineering Mathematics, 22nd Edition. S. Chand Publications.

Web Resources:

- https://en.wikipedia.org/wiki/System_of_linear_equations
- https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors
- https://www.math.hmc.edu/calculus/tutorials/eigenstuff/
- https://en.wikipedia.org/wiki/Quadratic_form
- https://en.wikipedia.org/wiki/Calculus
- https://en.wikipedia.org/wiki/Partial_derivative
- https://www.whitman.edu/mathematics/calculus_online/section14.03.html
- https://en.wikipedia.org/wiki/Multiple_integral
- http://tutorial.math.lamar.edu/Classes/CalcIII/MultipleIntegralsIntro.aspx

COURSE CODE	COMPUTATIONAL THINKING AND PROBLEM SOLVING USING C	Category ESC	L-T-P-C 3-0-0-3

Pre-requisite: Mathematics and logical reasoning at 10+2 level. No prior programming experience is assumed.

Course Objectives: The main objectives of the course are to

- Familiarize students with computers, computational thinking, and the process of algorithmic problem solving.
- Build strong programming fundamentals using the C language, including operators, control structures, and loops.
- Develop the ability to design and implement solutions using arrays, strings, and user- defined data types.
- Impart working knowledge of pointers, recursion, dynamic memory allocation, functions, and file handling for building modular, reusable programs.

Course Outcomes: *At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Apply computational thinking techniques (decomposition, abstraction, pattern recognition) to design algorithms and flowcharts for a given problem.	K3
CO2	Develop C programs using appropriate operators, expressions, selection statements, and loops to implement a given logic.	K3
CO3	Analyze problems involving arrays, strings, structures and unions and construct suitable C solutions.	K4
CO4	Apply pointers, recursion, and dynamic memory allocation to solve computational problems efficiently.	K3
CO5	Construct modular C programs using user-defined functions and perform text/binary file input-output operations.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	H	M	L	-	-	-	-	-	-	L	-	M	H
CO2	H	M	H	-	M	-	-	-	-	-	-	-	M	H
CO3	H	H	H	M	-	-	-	-	-	-	L	-	M	H
CO4	H	H	H	M	M	-	-	-	-	-	-	-	-	H
CO5	H	M	H	L	M	-	-	-	L	-	L	-	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Computational Thinking and Introduction to C** **Computational Thinking:** Decomposition, Pattern Recognition, Abstraction, Algorithm Design; representing solutions as algorithms, pseudocode, and flowcharts; Number Systems and Data Representation (integers, reals); Overview of Computer Organization and Program Execution.	12

	Introduction to C: Background, Structure of a C Program, Identifiers, Keywords, Variables, Constants, Basic Data Types, Input/Output; Expressions, Operators, type of operators- arithmetic, logical , relational, assignment, conditional, increment, decrement, etc., unary, binary, ternary, Operator Precedence and Associativity, Type Conversion; Scope, Storage Classes, and Type Qualifiers. **Bitwise Operators:** Logical Bitwise Operators, Shift Operators.	
UNIT – 2	**Selection and Decision-Making**: Relational and Logical Operators, Two-Way and Multi-Way Selection (if-else, switch), Nested Conditionals. **Repetition:** Pre-test and Post-test Loops, Counter- and Event-Controlled Loops, for/while/do-while, break and continue, Nested Loops; Programming Applications combining Selection and Repetition. **Arrays:** One-Dimensional and Two-Dimensional Arrays, Multidimensional Arrays, Array-based Problem Solving (searching, sorting, averaging).	9
UNIT – 3	**Strings:** String Representation, Standard String-Handling Functions, Arrays of Strings, String-to-Data Conversion. **User-Defined Types:** typedef, Enumerated Types, Structures, Unions, and Nested Structures; Programming Applications using Structures and Arrays of Structures. **Programming Applications**: Applications combining sections and repetitions	9
UNIT – 4	**Pointers:** Pointer Fundamentals, Pointer-to-Pointer, Pointer Arithmetic, Pointers and Arrays, Array of Pointers, Compatibility and l-value/r-value concepts. Dynamic Memory Allocation: malloc, calloc, realloc, free. **Functions:** Designing Structured/Modular Programs, Function Declaration and Definition, Parameter Passing (by value, by reference using pointers), Passing Arrays and Pointers to Functions, Standard Library Functions, Inter-Function Communication. **Recursion:** Recursive vs. Iterative Solutions, Designing Recursive Algorithms, Recursion with Arrays and Pointers, Efficiency Considerations.	9
UNIT – 5	**Preprocessor**: Directives and Macros. **File I/O:** Streams, Text vs. Binary Files, Standard Library File Functions, formatted and Character I/O, Sequential File Processing.	6
	Total	45

Text Books:

Forouzan, B. A., & Gilberg, R. F. (2006). Computer science: A structured programming approach using C (3rd ed.). Cengage Learning.

Kernighan, B. W., & Ritchie, D. M. (1988). The C programming language (2nd ed.). Prentice Hall India.

Reference Books:

1. Das, S. (2018). Computer fundamentals and programming (18th ed.). McGraw Hill Education.
2. Kamthane, A. N., & Kamthane, A. A. (2025). Programming in C (4th ed.). Pearson Education.
3. Dey, P., & Ghosh, M. (2013). Computer fundamentals and programming in C (2nd ed.). Oxford University Press.

Web References:

1. NPTEL – Programming in C, IIT Kanpur: https://onlinecourses.nptel.ac.in/noc21_cs81/preview
2. Geeks for Geeks – C Programming Language: https://www.geeksforgeeks.org/c-programming-language/
3. Programiz – Learn C Programming: https://www.programiz.com/c-programming

4. cplusplus.com – C Standard Library Reference: https://cplusplus.com/reference/clibrary/
5. TutorialsPoint – C Programming: https://www.tutorialspoint.com/cprogramming/index.html
6. W3Schools – C Tutorial: https://www.w3schools.com/c/

COURSE CODE	FUNDAMENTALS OF DIGITAL LOGIC	Category ESC	L-T-P-C 2-0-0-2

Pre-requisite: Basic Algebra, Set Theory Concepts, Binary Concepts and Logical Reasoning

Course Objectives: The main objectives of the course are to

- Develop proficiency in number systems, codes, and Boolean algebra as the mathematical basis of digital logic.
- Enable students to analyze and design combinational logic circuits using minimization techniques and standard building blocks.
- Introduce sequential logic elements such as latches and flip-flops and their role in storing digital state.
- Build the ability to design synchronous sequential circuits, including registers, counters, and finite state machines.

Course Outcomes :*At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Explain number systems, binary arithmetic, complements, binary codes, Boolean algebra, and the operation of basic and universal logic gates.	K2
CO2	Apply Boolean algebra and Karnaugh-map techniques to simplify Boolean expressions and design basic combinational circuits such as adders and subtractors.	K3
CO3	Analyze and design combinational logic circuits using multiplexers, demultiplexers, encoders, decoders, and magnitude comparators.	K4
CO4	Explain and analyze latches and flip-flops, including SR, D, JK, and T flip-flops, using excitation tables and timing diagrams.	K4
CO5	Design and evaluate sequential circuits using shift registers and asynchronous, synchronous, up/down, and modulo-N counters	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	**PO 1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PSO 1**	**PSO 2**	**PSO 3**
CO 1	H	M	L	-	-	-	-	-	-	-	-	M	-	L
CO 2	H	H	H	M	M	-	-	-	-	-	-	H	-	L
CO 3	H	H	H	M	M	-	-	-	-	-	-	H	-	M
CO 4	H	H	M	M	M	-	-	-	-	-	-	M	-	M
CO 5	H	H	H	H	H	-	-	-	-	-	M	H	-	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	No of Hours
UNIT – 1	Number Systems and Boolean Algebra Number systems (binary, octal, decimal, hexadecimal) and inter-conversions; binary arithmetic; complements (1's and 2's); binary codes (BCD, Gray code, ASCII); Boolean algebra postulates and theorems; logic gates (AND, OR, NOT, NAND, NOR, XOR, XNOR) and universal gates	7
UNIT – 2	Simplification of Boolean functions Minimization using Karnaugh maps (up to 4 variables); Sum-of-Products and Product-of-Sums forms; Combinational Logic Design I design of half-adder, full-adder, half-subtractor, full-subtractor	5
UNIT – 3	Combinational Logic Design II: multiplexers and demultiplexers; encoders and decoders; magnitude comparators. Sequential Logic Fundamentals- Latches and FlipFlops, Difference between combinational and sequential circuits;	5
UNIT – 4	Sequential Logic Design—-SR latch; clocked SR, D, JK, and T flip-flops; master-slave flip-flop; flip-flop excitation tables; timing diagrams; setup and hold time.	5
UNIT – 5	Registers, Counters, and Sequential Circuit Design Registers, Shift registers (SISO, SIPO, PISO, PIPO); ripple (asynchronous) counters; synchronous counters; up/down and modulo-N counters	5
Total		27

Text Books:

1. Mano, M. M., Ciletti, M. D. (2018). Digital design (6th ed.). Pearson.
2. Floyd, T. L. (2015). Digital fundamentals (11th ed.). Pearson.

Reference Books:

1. Hill, F. J., & Peterson, G. R. (2010). Introduction to switching theory and logic design (3rd ed.). Wiley.
2. Wakerly, J. F. (2018). Digital design: Principles and practices (5th ed.). Pearson.
3. Brown, S., &Vranesic, Z. (2014). Fundamentals of digital logic with VHDL design (3rd ed.). McGraw-Hill Education.

Web References:

1. NPTEL. (n.d.). Digital circuits and systems. National Programme on Technology Enhanced Learning. https://nptel.ac.in/
2. MIT OpenCourseWare. (n.d.). Digital systems laboratory. Massachusetts Institute of Technology. https://ocw.mit.edu/courses/6-111-introductory-digital-systems-laboratory-fall-2019/
3. All About Circuits. (n.d.). Digital circuits. https://www.allaboutcircuits.com/textbook/digital/

COURSE CODE	ENGLISH FOR TECHNICAL COMMUNICATION AND MOODLE LAB	Category HSM	L-T-P-C 0-0-3-1.5

Course Objectives:

- Develop active listening skills and the traits of an effective listener using lab activities and Ai-based listening tools.
- Develop speaking and presentation skills through structured practice and AI-speech coaching tools.
- Apply reading techniques and Analyze the texts to strengthen comprehension and employability.
- Apply writing techniques for technical and professional letters, emails, resumes and report writing to enhance employability.

Course Outcomes:

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Apply a variety of strategies to interpret and comprehend spoken texts/ discourse using contextual clues.	
CO2	Apply appropriate reading strategies to interpret content/material related to engineering and technology domain.	
CO3	Participate in discussions and presentations on general and technical topics.	
CO4	Possess an ability to write clearly on topics related to technology and workplace communication.	
CO5	Choose functional language, cohesive devices and skills of organization to express clearly in speaking.	
CO6	Demonstrate proficiency in using Moodle LMS and digital learning tools to access learning resources, manage academic activities, submit assignments, participate in quizzes and discussions, and organize learning effectively.	
CO7	Apply digital communication, academic, research, and presentation skills while demonstrating academic integrity, responsible use of AI and digital technologies, and professional online behaviour.	

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	–	M	–	–	L	M	–	M	M	H	–			
CO2	L	H	–	M	M	M	–	–	–	M	M			
CO3	–	M	–	–	M	M	–	H	H	H	M			
CO4	–	M	–	–	M	M	–	M	M	H	M			
CO5	–	M	–	–	L	M	–	H	H	H	M			
CO6	–	M	–	–	H	M	–	M	M	M	H			
CO7	–	M	L	M	H	M	M	H	H	H	H			

List of Experiments for the subject English for Technical Communication Lab

(Any 10 experiments in Part A)

1. **Phonetics & Accent Neutralisation**

- Sound Lab Drills — Students record minimal pairs (/p/–/b/, /t/–/d/) and compare waveforms using free apps like Voice Recorder.
- Pronunciation Clinics — Peer-correction circles where each student reads a short passage focusing on stress and intonation.
- IPA Treasure Hunt — Give IPA transcriptions; students identify the words and pronounce them.

- Shadowing Practice — Students mimic short clips (newsreader, documentary narrator) to reduce mother-tongue influence.

2. **Introducing Self and Others**
   - Speed Networking — Students rotate every 2 minutes and introduce themselves to a new partner.
   - Identity Cards — Each student prepares a "professional identity card" and introduces another student using it.
   - Role-based Introductions — Introduce yourself as a student, intern, job applicant, club member, etc.

3. **Brief Talk in Formal/Informal Contexts, JAM/Extempore**
   - JAM Box — A box with random topics; students pick one and speak for 60 seconds.
   - Context Switch — Students speak on the same topic twice: once formally (seminar), once informally (friends).
   - Prompt Cards — Situational prompts: "Convince your friend to join a club", "Address a class meeting".

4. **Listening Comprehension**
   - Audio Quizzes — Play short clips (announcements, interviews) and ask MCQs.
   - Gap-fill Listening — Provide partially blank transcripts; students fill missing words.
   - Error Spotting — Students listen to a narration and identify factual inconsistencies.

5. **Audio-based Note Taking & Narrating Events**
   - Event Reconstruction — Play an audio describing an incident; students reconstruct the event in 5–6 bullet points.
   - Timeline Creation — Students listen to a process or story and create a chronological timeline.
   - Peer Narration — One student listens to an audio; narrates it to a partner who writes notes.

6. **Process Description & Paragraph Writing (Technical Lab Context)**
   - Lab-to-Paragraph — After a lab experiment, students write a process description using sequence markers.
   - Flowchart to Paragraph — Provide a technical flowchart; students convert it into a coherent paragraph.
   - Reverse Engineering — Give a paragraph; students convert it back into steps and check accuracy.

7. **Group Discussions & Deliberations – Coming to Consensus**
   - Consensus Challenge — Groups must agree on one solution to a problem (e.g., "Best way to reduce campus waste").
   - Fishbowl GD — Inner circle discusses; outer circle observes and gives feedback.
   - Ranking Task — Groups rank items (e.g., skills for employability) and justify their consensus.

8. **Micro Presentations (2–5 minutes)**
   - One-Slide Presentation — Students prepare a single slide and speak for 3 minutes.
   - Teach a Concept — Students explain a simple technical idea (e.g., "How Wi-Fi works").
   - Object Talk — Students bring an object and present its use, history, or mechanism.

9. **Explaining Concepts – Technical Text-based**
   - Concept Simplification — Students take a technical paragraph and rewrite it for a layperson.
   - Peer Teaching — Each student teaches a concept from their discipline to a partner from another discipline.
   - Diagram Explanation — Provide a technical diagram; students explain it verbally.

10. **Presenting Logical Arguments – Debates**
- Two-Minute Debates — Quick debates with timed arguments and rebuttals.
- Evidence Hunt — Students gather data or examples to support their stance before debating.
- Role-based Debates — Students debate from assigned roles (environmentalist, policymaker, student).

11. **Talking About Social Issues & Data Interpretation**
- Data Talk — Provide charts/graphs (literacy rates, pollution levels); students interpret and present insights.
- Issue Circles — Groups discuss a social issue and propose one actionable solution.
- Poster-to-Speech — Students convert an infographic into a 2-minute talk.

12. **Interviews – Formal & Semi-formal**
- Mock Interview Panels — Students face a panel of peers acting as interviewers.
- Resume-based Q&A — Students ask each other questions based on their resumes.
- Situational Interviews — Semi-formal scenarios: club membership, internship, volunteer role.

Lab Manual
- Propel: A Course Book for Technical English Communication. Cengage Publishers, 2026.

Reference Books:
1. Raman Meenakshi,Sangeeta-Sharma.TechnicalCommunication.OxfordPress.2018.
2. Taylor Grant: English Conversation Practice,Tata McGraw Hill Education India,2016.
3. Hewing's, Martin. Cambridge Academic English (B2).CUP, 2012.
4. J.Sethi & P.V.Dhamija. A Course in Phonetics and Spoken English, (2ndEd), Kindle, 2013
5. Richards, Jack C., Jonathan Hull, and Susan Proctor. Interchange Level 3 Student's Book with Self-study DVD-ROM. Vol. 3. Cambridge University Press, 2012.

Web Resources:
1. www.esl-lab.com
2. www.englishmedialab.com
3. www.englishinteractive.net
4. https://www.britishcouncil.in/english/online
5. http://www.letstalkpodcast.com
6. https://www.youtube.com/c/mmmEnglish_Emma/featured
7. https://www.youtube.com/c/ArnelsEverydayEnglish/featured
8. https://www.youtube.com/c/engvidAdam/featured
9. https://www.youtube.com/c/EnglishClass101/featured
10. https://www.youtube.com/c/SpeakEnglishWithTiffani/playlists
11. https://www.youtube.com/channel/UCV1h_cBE0Drdx19qkTM0WNw
12. https://www.youtube.com/user/letstalkaccent/videos
13. https://www.youtube.com/c/EngLanguageClub/featured
14. https://www.youtube.com/channel/UC_OskgZBoS4dAnVUgJVexc
15. https://www.youtube.com/channel/UCNfm92h83W2i2ijc5Xwp_IA

Part B

(Any 4 Experiments in Part B)

Moodle Learning Management System: Digital learning an Academic Skills

Course Objectives

- To become familiar with model LMS and on-line learning
- To develop basic Digital and Academic Skills needed for Engineering students
- To learn effective on-line communication and collaboration
- To develop skills in Assignments, Quizzes, Presentation and Refreshing
- To practice Academic Integrity and Responsible use of Digital Technology

Moodle Learning Management System: Digital learning to enhance Academic Skills. Students familiarize with model LMS and on-line learning, communication and collaboration. They develop skills in Assignments, Quizzes, Presentation and online interviews. They exercise academic Integrity and responsible use of Digital Technology. They adopt the usage of Grammarly, ChatGPT, CultureWizard, Quizlet AI and Google Speech –to-text to enhance their respective skills.

List of Experiments

1. **Introduction to Moodle LMS**
   - Introduction to Moodle
   - Student login and profile management
   - Moodle dashboard and course navigation
   - Accessing announcements and course materials
   - Notifications and calendars

2. Learning Through Moodle

- Accessing PDFs, videos, web resources, and lecture materials
- Downloading and organizing study materials
- Moodle lessons and learning activities
- Tracking course completion
- Managing deadlines and study schedules

3. Assignments, Quizzes and Assessments

- Understanding assignment instructions
- Creating and uploading assignments
- File formats and file naming
- Online quizzes and question types
- Viewing marks and teacher feedback
- Resubmission and revision

4. Digital Communication and Collaboration

- Moodle forums and discussions
- Academic email etiquette
- Online communication and netiquette
- Group activities and peer learning
- Collaborative learning tools
- Digital presentation skills

5. Academic Skills for Engineering Students

- Effective note-taking

- Reading and understanding technical material
- Time management and study planning
- Problem-solving and critical thinking
- Basic research and information-search skills
- Preparing reports and presentations
- Introduction to citations and referencing

6. Academic honesty

- Plagiarism and how to avoid it
- Copyright and fair use
- Responsible use of AI tools
- Digital privacy and cybersecurity
- Evaluating reliable online information
- Professional behavior in online environments
- Practical Activities

7. Students can complete activities such as:

- Create and update a Moodle profile.
- Download course material and organize it into folders.
- Submit an assignment through Moodle.
- Attempt an online quiz.
- Participate in a Moodle discussion forum.
- Upload a revised assignment after receiving feedback.
- Prepare a short technical presentation.

COURSE CODE	APPLIED CHEMISTRY FOR EMERGING TECHNOLOGIES LAB	Category BSC	L-T-P-C 0-0-3-1.5

Pre-requisite: NILs

Course Objectives:

- To provide hands-on experience in determining fundamental constants and semiconductor band-gap energy through optical and photoelectric experiments.
- To train students in conductometric, potentiometric and cell-based electrochemical measurement techniques.
- To familiarize students with the performance evaluation of batteries, supercapacitors and biosensors.
- To develop skills in the preparation and characterisation of nanomaterials and in safe laboratory practice, data analysis and technical reporting.

Course Outcomes (COs): ***At the end of the course, students will be able to***

CO	Course Outcome	Knowledge Level (K)#
CO1	Estimate fundamental physical quantities such as Planck's constant and the band-gap energy of a semiconductor from photoelectric and optical absorption measurements.	K3
CO2	Determine electrochemical parameters including EMF, specific and equivalent conductance, and pH using conductometric, potentiometric and cell-based methods.	K3
CO3	Analyse the charge-discharge characteristics and performance of batteries and supercapacitors used in computing devices.	K4
CO4	Synthesize and examine nanomaterials and evaluate their optical and adsorption properties for engineering applications.	K4

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	M	L	H	M	-	-	M	L	-	M			
CO2	H	M	-	H	M	-	-	M	L	-	M			
CO3	H	H	M	H	M	-	-	M	L	-	M			
CO4	H	H	M	H	M	M	-	M	L	-	M			

LIST OF EXPERIMENTS

(Any **TEN** of the following listed experiments)
(Out of which any **TWO** experiments may be conducted in virtual mode)

1. Verification of the photoelectric effect and estimation of Planck's constant.

2. Study of the emission spectrum of a hydrogen/mercury-vapour Lamp using a spectrometer.

3. Determination of the band-gap energy of a semiconductor sample from its optical absorption edge.

4. Determination of a strong acid against a strong base by using a conductometer.

5. Determination of the EMF of a Daniel cell and verification of the Nernst equation.

6. Determination of the charge–discharge characteristics of a lithium-ion battery/supercapacitor cell.

7. Determination of the specific and equivalent conductance of an electrolyte solution.

8. Assemble a simple Zn-carbon (or lead-acid) cell and measure its voltage and approximate capacity.

9. Construction and performance evaluation of a simple enzyme based electrochemical biosensor for glucose detection (demo)

10. Preparation and observation of colloidal gold/silver nanoparticles (size-dependent colour- change)

11. Determination of the viscosity of a polymer/nanomaterial-dispersion solution using an Ostwald viscometer.

12. Determination of the pH of an electronic-grade (deionized) water sample used in wafer/PCB cleaning.

13. Determination of strong acid against a strong base by using potentiometer.

14. Determination of conductivity of an electronic-grade (deionized) water sample used in wafer/PCB cleaning.

15. Redox titration by using potentiometer.

16. Determination of strength of an acid in Pb-Acid battery.

17. Adsorption capacity of a nanomaterial (e.g., activated carbon nanoparticles or nano-iron oxide) for removal of a dye or metal ion from water.

18. Study of galvanic corrosion using a zinc–iron electrochemical couple.

References:

- Vogel, A. I. (2012). Vogel's textbook of quantitative chemical analysis (6th ed.). Pearson Education.
- Ahluwalia, V. K., & Dhingra, S. (2017). Comprehensive practical organic chemistry. Universities Press.
- Virtual Labs, Ministry of Education, Government of India. https://vlab.co.in

COURSE CODE	COMPUTATIONAL THINKING AND PROBLEM SOLVING USING C LAB	Category ESC	L-T-P-C 0-0-3-1.5

Pre-requisite: - Basic understanding of computer fundamentals and problem-solving concepts

Course Objectives: The main objectives of the course are to

- Apply the principles of C language and computational thinking techniques in problem solving.s
- Design flowcharts and algorithms, and to develop the ability to trace and debug programs.
- Design and develop C programs using arrays, strings, pointers, recursion, and functions.
- Apply dynamic memory allocation, file operations, and preprocessor commands in program design.

Course Outcomes: *At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Apply computational thinking to construct algorithms and flowcharts for a given problem and translate them into working C programs.	K3
CO2	Develop C programs using operators, selection statements and loops to implement a specified logic.	K3
CO3	Develop C programs that manipulate arrays, strings, structures and pointers to solve computational problems.	K3
CO4	Construct modular programs using user-defined functions and recursion, and perform file input-output operations.	K3
CO5	Analyse a C program by tracing and debugging it, and evaluate alternative implementations for correctness and efficiency.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	H	M	L	H	-	-	L	L	-	M	-	M	H
CO2	H	M	H	-	H	-	-	L	L	-	M	-	M	H
CO3	H	H	H	M	H	-	-	L	L	-	M	-	M	H
CO4	H	H	H	M	H	-	-	L	L	-	M	-	M	H
CO5	H	H	M	M	H	-	-	L	L	-	H	-	M	H

(Please fill the above with Levels of Correlation, viz., L, M, H)

Lab Requirements (tools / software to carry out the experiments):

- C compiler: GCC (GNU Compiler Collection) — via MinGW-w64 on Windows, or the built-in gcc on Linux/macOS.
- IDE / editor: Code::Blocks, Dev-C++ or Visual Studio Code with the C/C++ extension.
- Command-line terminal to compile and run programs (gcc, ./a.out) and observe the compile–link–run stages.
- Debugger: GDB (GNU Debugger), integrated in Code::Blocks / VS Code, for tracing and step-by-step debugging.
- Operating system: Windows 10/11 or a Linux distribution (e.g., Ubuntu); any 64-bit PC with 4 GB RAM or more.
- No-install alternative: OnlineGDB, Programiz or Replit for quick practice and testing in a browser.

List of Experiments

Exercise 1: Getting Started with

1. Write, compile and run a C program that prints "Hello, World!" on the screen. Identify the source file, the object file and the executable produced, and record the command used at each stage.
2. Write a C program that declares variables of type int, float, char and double, reads a value for each from the user, and displays each value together with the size of its type using sizeof.
3. Write a C program to read the height and width of a rectangle from the user and compute its perimeter and are.

Exercise 2: Operators and Expressions

1. Write a C program to compute the distance between two points whose coordinates are entered by the user.
2. Write a C program that reads four integers p, q, r, s and, using relational and logical operators, prints 'Correct values' if q > r, s > p and (r + s) > (p + q); otherwise 'Wrong values'.
3. Write a C program to compute simple and compound interest for values entered by the user.

Exercise 3: Selection and Decision Making

1. Write a C program to read a character and classify it as a vowel, consonant, digit or special symbol.
2. Write a C program to find the largest of three numbers using nested if-else.
3. Write a menu-driven C program to compute the area of various geometrical shapes (circle, rectangle, triangle, square) using switch-case.

Exercise 4: Loops and Series

1. Write a C program to display the first n even natural numbers and their sum.
2. Write a C program to reverse a given integer and check whether it is a palindrome.
3. Write a C program to display the sum of the harmonic series 1 + 1/2 + 1/3 + ... up to n terms.

Exercise 5: Nested Loops and Number Patterns

1. Write a C program to print the multiplication table pyramid for numbers 1 to n using nested loops.
2. Write a C program to check whether a given number is prime, then extend it to print all prime numbers and all Armstrong numbers below.

Exercise 6: One-Dimensional Arrays

1. Write a C program to print all unique elements of an array, and implement linear search to find a given element in the array.
2. Write a C program to separate the odd and even elements of an array into two different arrays, and implement binary search to find a given element in a sorted array.
3. Write a C program to sort the elements of an array in ascending order using selection sort, and compare the efficiency of linear search vs. binary search on the sorted array by counting the number of comparisons each method takes.

Exercise 7: Matrices

1. Write a C program to perform addition of two square matrices and display the resultant matrices.
2. Write a C program to multiply two square matrices and display the resultant matrices.
3. Write a C program to find the transpose of a given matrix.

Exercise 8: Strings:

Write C programs to perform the following without using string handling functions

- **Find the length** of a given string.
- **Copy** one string into another.

- **Concatenate** two strings.
- **Compare** two strings and display whether they are equal, or which one is greater.
- **Reverse** a given string (using a separate swap function to exchange characters).
- **Search for a character** in a string and display its position (or indicate if not found).
- **Count the number of vowels, consonants, and words** in a given string

Exercise 9: Pointers Fundamentals

1. Write a C program to demonstrate the use of the & (address-of) and * (value-at-address) operators.
2. Write a C program to add two numbers using pointers.
3. Write a C program to swap two values using call by reference.**Exercise 10: Pointers and Dynamic Memory**
    1. Write a C program to dynamically allocate memory for an array using malloc, find the largest element and the sum of all elements in it, then repeat using calloc and note the difference between the two.
    2. Write a C program to count the number of vowels in a string using a pointer.

Exercise 10: User-Defined Functions

1. Write a C program to check whether a number is prime using a user-defined function.
2. Write a C program to compute the GCD and LCM of two numbers using functions.
3. Write a C program to demonstrate a function that returns a pointer.

Exercise 11: Recursion

1. Write a C program to compute the factorial and the n-th Fibonacci term using recursion.
2. Write a C program to compute the sum of digits of a number using recursion.
3. Write a C program to convert a decimal number to its binary equivalent using a recursive function.

Exercise 12: File Handling

1. Write a C program to append multiple lines at the end of an existing text file.
2. Write a C program to copy the contents of one file into another file.
3. Write a C program to count the number of characters, words and lines in a text file.

Web References:

- NPTEL – Programming in C, IIT Kanpur: https://onlinecourses.nptel.ac.in/noc21_cs81/preview
- Virtual Labs – Problem Solving Lab, IIIT Hyderabad: https://ps-iiith.vlabs.ac.in/
- GeeksforGeeks – C Programming: https://www.geeksforgeeks.org/c-programming-language/

COURSE CODE	IT WORKSHOP	Category ESC	L-T-P-C 0-0-3-1.5

Pre-requisite: Familiarity with basic computer operation is helpful but not assumed.

Course Objectives: The main Objectives of the course are to

- Familiarize students with the internal components of a computer and enable them to assemble/disassemble a PC and install operating systems.
- Introduce students to virtualization and cloud computing as modern approaches to working with multiple operating systems and remote infrastructure.
- Enable students to connect to and navigate the Internet safely, and to understand netiquette and cyber hygiene practices.
- Develop proficiency in office automation tools (Word, Excel, PowerPoint) for academic and professional documentation, data analysis, and presentation**.**

Course Outcomes: *At the end of the course, student will be able to*

CO	Course Outcome Statement	Knowledge Level (K)#
CO1	Identify computer hardware components and assemble, disassemble and set up a working PC with the required operating system.	K3
CO2	Connecting to the Network or remote computing environment.	K3
CO3	Apply safe browsing, effective search and cyber hygiene practices while working on the Internet.	K3
CO4	Create formatted technical documents such as certificates, abstracts and newsletters using Word	K3
CO5	Build spreadsheets with formulas, functions and charts for data analysis, and design structured interactive presentations.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	M	L	M	-	H	-	-	L	L	-	M	L	L	L
CO2	M	L	M	-	H	-	-	L	L	-	H	M	L	M
CO3	M	M	L	-	H	L	M	L	L	-	H	L	L	H
CO4	L	-	L	-	H	-	-	L	H	M	M	L	-	-
CO5	L	M	M	L	H	-	-	L	H	M	M	M	M	-

(Please fill the above with Levels of Correlation, viz., L, M, H)

Lab Requirements (tools / software to carry out the experiments):

- A working desktop PC with peripherals that can be opened, an antistatic wrist strap and a basic screwdriver toolkit.
- OS installation media: Windows 10/11
- Office suite: MS Office (Word, Excel, PowerPoint)
- Internet with a modern browser (Chrome / Firefox / Edge), useful extensions, and antivirus / 2FA for the cyber-hygiene tasks.

List of Experiments

PC Hardware and Systems

Exercise 1: Identifying Components
Identify the peripherals of a computer and the components inside the CPU cabinet (motherboard, processor, RAM, SMPS, storage drives such as HDD/SSD/NVMe, and graphics card) and their functions. Draw the block diagram of the CPU with the configuration of each component.

Exercise 2: Assembling a PC
Disassemble a PC and reassemble it back to working condition. The lab instructor verifies the work and conducts a viva; refer to the demonstration video provided with the course content.

Exercise 3: Operating System Installation
Individually install Windows on a personal computer. The lab instructor verifies the installation and conducts a viva.

Internet and World Wide Web

Exercise 4: Network Connectivity
Connect the system to a LAN/Wi-Fi and access the Internet by configuring TCP/IP settings (IP address, DNS). Demonstrate access to websites and email, and use basic troubleshooting commands (ping, ipconfig/ifconfig, tracert).

Exercise 5: Browsers, Search and Netiquette
Customize a web browser with bookmarks, pop-up blockers and useful extensions (PDF viewer, ad blocker, password manager). Use search engines effectively with filters and Boolean operators, and demonstrate proper netiquette for online communication.

Exercise 6: Cyber Hygiene
Identify common online threats (phishing, malware, unsafe downloads) and configure the system for safe browsing: HTTPS-only browsing, strong passwords with a password manager, two-factor authentication, and keeping the OS and antivirus updated.

Word Processing (Word)

Exercise 7: Certificate Creation
Using MS Word, create a project certificate covering fonts, drop cap, text effects, character spacing, borders and colours, headers and footers, and date/time options.

Exercise 8: Project Abstract
Create a project document on a topic of your choice, and demonstrate the use of formatting styles, tables, bullets and numbering, text direction, cell alignment, footnotes, hyperlinks, symbols, spell check, and track changes.

Exercise 9: Newsletter, and Introduction
Create a newsletter covering table of contents, newspaper columns, images and clip art, the drawing toolbar and WordArt, image formatting, text boxes, paragraph formatting and mail merge.

Spreadsheets and Presentations

Exercise 10: Spreadsheet Scheduler and GPA

Using MS Excel , create a scheduler (gridlines, cell formatting, AutoFill, summation) and a GPA sheet using cell referencing, the AVERAGE, STDEV and COUNT functions, charts, and multiple worksheets.

Exercise 11: Spreadsheet Data Analysis

Perform data analysis using LOOKUP/VLOOKUP, splitting cells, freezing panes, grouping/outlining, sorting, Boolean and logical operators, conditional formatting and a basic pivot table.

Exercise 12: Interactive Presentation

Create an interactive presentation covering slide layouts, WordArt, bullets and autoshapes; hyperlinks; inserting images, audio, video, tables and charts; master layouts (slide, template, notes); views; and backgrounds and design templates.

Web References:

- Virtual Labs – Computer Organisation and Hardware: https://www.vlab.co.in/broad-area-computer-science-and-engineering
- AWS Free Tier: https://aws.amazon.com/free/

COURSE CODE	SPORTS AND YOGA OR NSS/NCC	Category MNC	L-T-P-C 0-0-2-0

Course Objectives:

The objective of introducing this course is to impart discipline, character, fraternity, teamwork, social consciousness among the students and engaging them in selfless service.

Course Outcomes: *After completion of the course the students will be able to*

CO	Course Outcome	Knowledge Level (K)#
CO1	Understand the importance of discipline, character and service motto.	
CO2	Solve some societal issues by applying acquired knowledge, facts, and techniques.	
CO3	Explore human relationships by analyzing social problems.	
CO4	Determine to extend their help for the fellow beings and downtrodden people.	
CO5	Develop leadership skills and civic responsibilities.	

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1														
CO2														
CO3														
CO4														
CO5														

UNIT	CONTENTS	Contact Hours
UNIT - 1	Orientation General Orientation on NSS/NCC/ Scouts & Guides/Community Service activities, career guidance. Activities: Conducting –ice breaking sessions-expectations from the course-knowing personal talents and skills Conducting orientations programs for the students –future plans-activities-releasing road map etc. Displaying success stories-motivational biopics- award winning movies on societal issues etc. Conducting talent show in singing patriotic songs-paintings-any other contribution.	
UNIT - 2	Nature & Care Activities: Best out of waste competition. Poster and signs making competition to spread environmental awareness. Recycling and environmental pollution article writing competition. Organising Zero-waste day. Digital Environmental awareness activity via various social media platforms. Virtual demonstration of different eco-friendly approaches for sustainable living. Write a summary on any book related to environmental issues.	
UNIT - 3	Community Service Activities: Conducting One Day Special Camp in a village contacting village-area leaders- Survey in the village, identification of problems-helping them to solve via media- authorities-experts-etc. Conducting awareness programs on Health-related issues such as General Health, Mental health, Spiritual Health, HIV/AIDS, Conducting consumer Awareness. Explaining various legal provisions etc. Women Empowerment Programmes- Sexual Abuse, Adolescent	

UNIT	CONTENTS	Contact Hours
	Health and Population Education. Any other programmes in collaboration with local charities, NGOs etc.	

Reference Books:

1. Nirmalya Kumar Sinha & Surajit Majumder, *A Text Book of National Service Scheme* Vol; I, Vidya Kutir Publication, 2021 (ISBN 978-81-952368-8-6)
2. *Red Book - National Cadet Corps* – Standing Instructions Vol I & II, Directorate General of NCC, Ministry of Defence, New Delhi
3. Davis M. L. and Cornwell D. A., "Introduction to Environmental Engineering", McGraw Hill, New York 4/e 2008
4. Masters G. M., Joseph K. and Nagendran R. "Introduction to Environmental Engineering and Science", Pearson Education, New Delhi. 2/e 2007
5. Ram Ahuja. *Social Problems in India*, Rawat Publications, New Delhi.

General Guidelines:

1. Institutes must assign slots in the Timetable for the activities.
2. Institutes are required to provide instructor to mentor the students.

Evaluation Guidelines:

- Evaluated for a total of 100 marks.
- A student can select 6 activities of his/her choice with a minimum of 01 activity per unit. Each activity shall be evaluated by the concerned teacher for 15 marks, totalling to 90 marks.
- A student shall be evaluated by the concerned teacher for 10 marks by conducting viva voce on the subject.

COURSE CODE	PHYSICS FOR ADVANCED COMPUTING	Category BSC	L-T-P-C 3-0-0-3

Course Objectives:

1. To equip students the concepts of interference, diffraction and polarization and their applications in optics and communication systems.
2. To understand the basic concepts of Laser light sources and working principle of optical fibers and identify the engineering applications of lasers and optical fibers in communication.
3. To explain various electron theories and interpret the effect of temperature on Fermi-Dirac distribution function and provide basic knowledge of quantum mechanics for quantum computing applications.
4. To familiarize students with semiconductors and magnetic materials and their applications.
5. To impart knowledge on low temperature physics and superconductivity and their applications.

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Apply the principles of interference, diffraction and polarization to analyze optical systems and engineering applications.	K3
CO2	Analyze the operation of various laser systems as well as the working principle of optical fibers in relation to communication applications.	K4
CO3	Interpret the effects of temperature on Fermi energy and Apply the concepts of quantum mechanics to explain modern computational systems like Quantum Computing.	K3
CO4	Analyse semiconductor characteristics and magnetic material behaviour for engineering and electronic applications.	K4
CO5	Apply the concepts of low-temperature physics and superconductivity in modern technological applications and intelligent systems.	K3

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	M	–	–	M	–	–	–	–	L	M			
CO2	H	M	M	L	H	–	–	–	L	L	M			
CO3	H	H	M	M	H	–	–	–	L	L	M			
CO4	M	H	M	M	M	L	–	–	L	L	M			

UNIT	CONTENTS	Contact Hours
UNIT - 1	APPLIED WAVE OPTICS Interference: Superposition Principle – Coherence – Types of Interference – Thin-film interference (reflection geometry) – Newton's rings – Determination of wavelength and refractive index. Diffraction and Optical Surveying: Fresnel and Fraunhofer diffraction – Fraunhofer's diffraction due to single slit, double slit & N-slits (qualitative) – Diffraction Grating and grating spectrum – Dispersive and Resolving power of	10

UNIT	CONTENTS	Contact Hours
	a grating. Polarization: Types of polarization – Nicol's prism – Half-wave and Quarter-wave plates.	
UNIT - 2	QUANTUM PHOTONICS Lasers: Principles of Absorption – Spontaneous and Stimulated Emission – Population Inversion – Three level and Four level laser systems – Construction and working of Ruby and He-Ne Laser – Applications (qualitative). Fiber Optics: Propagation mechanism – Numerical Aperture – Acceptance angle – Types of optical fibres – Classifications and Modes of optical fibres.	9
UNIT - 3	FREE ELECTRON THEORY & QUANTUM MECHANICS Classical free electron theory – merits and demerits – Quantum free electron theory – equation for electrical conductivity – Fermi-Dirac statistics – Fermi level and variation with temperature. Quantum Mechanics: de Broglie concept of matter waves – Heisenberg's uncertainty principle – Wave function – Schrödinger's time dependent and time-independent wave equations – Particle in a one-dimensional potential well – Quantum tunnelling (basic idea).	10
UNIT - 4	SEMICONDUCTORS & MAGNETIC MATERIALS Semiconductors: Formation of energy bands – Classification of solids - Intrinsic semiconductors – Density of charge carriers – Electrical conductivity – Fermi level – extrinsic semiconductors: density of charge carriers (qualitative) – dependence of Fermi energy on carrier concentration and temperature – Drift and diffusion currents – Einstein's equation – Hall effect – Hall coefficient – Applications of Hall effect. Magnetic Materials: Introduction and classification - Dia, Para, Ferro, Ferri, and Anti-ferro; Hysteresis and Eddy current losses in electrical machines – Skin effect; Soft & Hard magnetic materials.	10
UNIT - 5	LOW TEMPERATURE PHYSICS & SUPERCONDUCTIVITY Low Temperature Physics: 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). Superconductors: Properties – Critical parameters – Meissner effect – Type-I and Type-II superconductors – BCS theory – Cooper pairs and energy gap – DC and AC Josephson effect – Applications (qualitative only): SQUIDs, magnetic levitation and lossless power transmission.	9
	Total	**48**

Textbooks

1. Avadhanulu, M. N., Kshirsagar, P. G., & Murthy, T. V. S. A. (2019). A textbook of engineering physics (11th ed.). S. Chand Publishing.
2. Palanisamy, P. K. (2019). Engineering physics. SciTech Publications.

Reference Books

1. Bhattacharya, D. K., & Tandon, P. (2015). Engineering physics. Oxford University Press.
2. Nielsen, M. A., & Chuang, I. L. (2010). Quantum computation and quantum information (10th anniversary ed.). Cambridge University Press.

Learning Resources & NPTEL Links

- NPTEL – nptel.ac.in

- MIT OpenCourseWare – ocw.mit.edu
- Nave, C. R. (n.d.). HyperPhysics. Georgia State University. http://hyperphysics.phy-astr.gsu.edu/
- University of Colorado Boulder. (n.d.). PhET interactive simulations. https://phet.colorado.edu/
- Library of Congress. (n.d.). Science and technology: Selected internet resources – Physics. https://www.loc.gov/rr/scitech/selected-internet/physics.html

COURSE CODE	DIFFERENTIAL EQUATIONS AND VECTOR CALCULUS	Category BSC	L-T-P-C 3-0-0-3

Pre-requisite: Mathematics of 10+2 / Intermediate level (algebra, matrices and basic calculus)

Course Objectives:

- This course is aimed to provide the learner with several methods for solving differential equations and partial differential equations.
- To prepare the student to calculate gradient, surface and volume integrals by applying standard theorems.

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Model simple physical problems involving rate as first order differential equations and solve the resulting equation	K2, K3, K5
CO2	Apply differential equation techniques to formulate and solve homogeneous and non-homogeneous differential equations, solve simultaneous linear differential equations, and apply these methods to model and analyze L-C-R circuit problems and simple harmonic motion	K2, K3, K5
CO3	Solve partial differential equations that model physical processes	K2, K3, K5
CO4	Apply the knowledge of gradient, divergence and curl and interpret the physical meaning of different operators	K2, K3, K5
CO5	Compute the work done against a field, circulation and flux using vector calculus methods	K2, K3, K5

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	H	M											
CO2	H	H	M											
CO3	H	H	M											
Co4	H	H	M											
CO5	H	H	M											

UNIT	CONTENTS	Contact Hours
UNIT - 1	Differential Equations of First Order and First Degree: Exact equations and equations reducible to exact form (integrating factor), Linear differential equations – Bernoulli's equations. Orthogonal Trajectories, Applications: Newton's Law of cooling – Law of natural growth and decay- Electrical circuits.	
UNIT - 2	Linear Differential Equations of Higher Order (Constant Coefficients): Definitions, Homogeneous and non-homogeneous, Complementary function, Particular integral, General solution, Method of variation of parameters-Wronskian. Simultaneous linear equations, Applications to L-C-R Circuit problems and Simple Harmonic motion.	
UNIT - 3	Partial Differential Equations: Introduction and formation of Partial Differential Equations by elimination of arbitrary constants and arbitrary functions, Solutions of first order linear equations using Lagrange's method, Non-linear partial	

UNIT	CONTENTS	Contact Hours
	differential equations using Charpit's Method. Homogeneous Linear Partial differential equations with constant coefficients.	
UNIT - 4	Vector Differentiation: Scalar and Vector point functions, Vector operator Del, Gradient and applications, Directional derivative, Divergence-Solenoidal and Curl-Irrotational Field - Scalar Potential, Vector identities.	
UNIT - 5	Vector Integration: Line integral – Circulation – Work done by the force, Surface integral-flux, Green's theorem in a plane (without proof), Stoke's theorem (without proof), Volume integral, Divergence theorem (without proof) and their applications.	

Learning Resources:

Text Books:

3. Grewal, B.S. (2024). Higher Engineering Mathematics, 45th Edition. Khanna Publishers.
4. Kreyszig, E. (2025). Advanced Engineering Mathematics, 11th Edition. John Wiley & Sons.

Reference Books:

4. Ramana, B.V. (2017). Higher Engineering Mathematics. McGraw Hill Education.
5. James, G. (2018). Advanced Modern Engineering Mathematics. 5th Edition Pearson publishers.
6. Dass, H.K. (2022). Advanced Engineering Mathematics, 22nd Edition. S. Chand Publications.

Web Resources:

- https://mathworld.wolfram.com/First-OrderOrdinaryDifferentialEquation.html
- https://en.wikipedia.org/wiki/Differential_equation
- https://en.wikipedia.org/wiki/Partial_differential_equation
- https://en.wikipedia.org/wiki/Vector_calculus
- https://en.wikipedia.org/wiki/Vector_calculus

COURSE CODE	DATA STRUCTURES USING C	Category ESC	L-T-P-C 3-0-0-3

Pre-requisite: Computational Thinking and Problem Solving using C

Course Objectives: At the end of the course, the student should be able to:

- Understand the fundamental concepts of algorithms, structured programming, and abstract data types using the C programming language.
- Apply programming constructs and basic data structures to organize and manipulate data efficiently.
- Develop problem-solving skills by implementing linear and non-linear data structures using C.
- Analyze the performance of searching, sorting, and traversal algorithms using basic complexity measures.
- Apply appropriate data structures to solve computational problems encountered in engineering applications.

Course Outcomes: *At the end of the course, student will be able to*

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Explain fundamental data structure concepts, abstract data types, classifications, and analyze the time and space complexity of algorithms	K2
CO2	Implement and evaluate searching and sorting algorithms, including linear search, binary search, bubble sort, selection sort, insertion sort, merge sort, and quick sort, based on their computational efficiency	K3
CO3	Design and implement linear data structures such as linked lists, stacks, and queues using arrays, pointers, and dynamic memory allocation, and apply them to solve computational problems.	K4
CO4	Implement and apply tree data structures, including binary trees, binary search trees, and heaps, using appropriate traversal and manipulation techniques.	K3
CO5	Represent and process graphs and hash tables using appropriate techniques, and implement graph traversal and minimum spanning tree algorithms along with hashing and collision-resolution methods.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO 3
CO1	H	M	-	M	M	-	-	-	L	-	M	H	M	L
CO2	H	H	M	M	H	-	-	-	L	-	M	H	M	L
CO3	H	H	H	M	H	-	-	M	L	L	M	H	M	L
CO4	H	H	H	M	H	-	-	M	L	L	M	H	H	L
CO5	H	H	H	M	H	L	-	M	L	L	H	H	H	M

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	Introduction to data Structures, Abstract Data Types (ADT); Need and classification of Data Structures; Introduction to algorithm analysis; Time and space complexity (basic concepts). **Searching and Sorting:** Linear Search; Binary Search; Bubble Sort; Selection Sort; Insertion Sort; Merge Sort; Quick Sort;	9
UNIT – 2	**Linear Data Structures:** Dynamic memory allocation; Recursion; Linked Lists—Singly, Doubly, and Circular Linked Lists; Basic operations (creation, insertion, deletion, traversal); Applications of Linked Lists;	9
UNIT – 3	**Stacks**—Array and Linked List implementations; operations on stacks, Basic applications – infix to postfix conversion, postfix expression evaluation, parenthesis matching; **Queues**—Simple Queue, Circular Queue, operations and Applications of Queues.	9
UNIT – 4	**Trees:** Introduction and terminology; Binary Trees; Tree representation; Tree traversals; Binary Search Trees—Insertion, Deletion, Searching; Heap—Min Heap and Max Heap; Applications of Trees.	9
UNIT – 5	**Graphs:** Graph terminology; Graph representations (Adjacency Matrix and Adjacency List); Graph traversals—Breadth First Search (BFS) and Depth First Search (DFS); Introduction to Minimum Spanning Trees (Prim's Algorithm); Applications of Graphs. **Hashing:** Introduction to Hashing; Hash Functions; Collision Resolution , open addressing, linear probing, quadratic probing,, double hashing, separate chaining, rehashing, examples	9
	Total	**45**

Text Books:

- Balagurusamy, E. (2025). Data Structures Using C (2nd ed.). McGraw Hill Education India.
- Thareja, R. (2023). Data Structures Using C (3rd ed.). Oxford University Press India.

Reference Books:

- Deshpande, P. S., & Kakde, O. G. (2003). C and data structures. Dreamtech Press.
- Shukla, R. K. (2009). Data structures using C and C++. Wiley India.

Web Resources:

- National Programme on Technology Enhanced Learning. (2026). Introduction to Programming in C. IIT Kanpur. https://nptel.ac.in/courses/106104128
- National Programme on Technology Enhanced Learning. (2026). Programming, Data Structures and Algorithms using C. IIT Madras. https://nptel.ac.in/courses/106106127
- National Programme on Technology Enhanced Learning. (2026). Problem Solving Through Programming in C. IIT Kharagpur. https://onlinecourses.nptel.ac.in
- Virtual Labs: https://ds1-iiith.vlabs.ac.in/
- Programiz. Data Structures. https://www.programiz.com/dsa
- Visualgo. Data Structure and Algorithm Visualizations. https://visualgo.net/en

COURSE CODE	PYTHON PROGRAMMING	Category ESC	L-T-P-C 2-0-0-2

Pre-requisite: Problem Solving and Programming Using C.

Course Objectives: The main objectives of the course are to

- Develop proficiency in Python control structures while explicitly linking conditionals to abstraction and loops to pattern recognition.
- Build competence in selecting Python data structures (strings, lists, tuples, dictionaries)
- Introduce object-oriented programming concepts such as classes, objects, inheritance, and polymorphism, enabling students to model real-world problems through encapsulation and abstraction.
- Train students to decompose problems into functions and recursive sub-problems, use lambda expressions and functional tools, reason informally about algorithmic efficiency, and use AI coding assistants responsibly.
- Equip students with file handling, exception handling, and basic data analysis/visualisation skills, framed as decomposing a larger data workflow into manageable, robust steps and to integrate these skills in a capstone mini-project.

Course Outcomes: *At the end of the course, student will be able to*

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Develop Python programs using variables, data types, operators, input/output, conditional statements, loops, and appropriate coding practices to solve computational problems.	K3
CO2	Implement Python programs using strings, lists, tuples, dictionaries, and comprehensions to efficiently represent, manipulate, and process different types of data.	K3
CO3	Develop and analyze modular Python programs using functions, lambda expressions, functional programming tools, recursion, modules, and libraries, considering basic algorithmic efficiency.	K4
CO4	Design and implement object-oriented Python programs using classes, objects, encapsulation, inheritance, polymorphism, operator overloading, and composition to model real-world problems.	K4
CO5	Develop Python applications that perform file processing and exception handling, and apply NumPy, pandas, and Matplotlib for basic data analysis and visualization.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	M	M	-	H	-	-	L	L	-	M	H	M	L
CO2	H	H	H	M	H	-	-	L	L	-	M	H	H	L
CO3	H	H	H	M	H	-	-	M	L	-	H	H	H	M
CO4	H	H	H	M	H	-	-	M	M	L	H	H	M	M
CO5	H	H	H	H	H	M	-	M	M	L	H	H	H	M

UNIT	CONTENTS	Contact Hours
UNIT – 1	**PYTHON BASICS AND CONTROL FLOW** **Python Basics:** Translating algorithms and flowcharts into Python, variables, data types, operators and input/output. Good Coding Practice: Docstrings, inline comments and print-based debugging (threaded through every unit from here on). **Conditional Statements:** if, if-else and chained/nested conditionals, abstraction over multiple cases. **Loops:** for and while loops, pattern recognition of repetition. **Loop Control:** Nested loops and loop control (break, continue, pass), decomposition of complex repetitive logic.	9
UNIT – 2	**DATA STRUCTURES IN PYTHON** **Strings**: Pattern matching and abstraction of text as a sequence; string operations and methods. **Lists:** Abstraction of ordered, mutable collections; list operations; list comprehensions. **Tuples and Dictionaries:** Abstraction of fixed (tuples) and key-based (dictionaries) data; common operations and use-case selection.	9
UNIT – 3	**FUNCTIONS, RECURSION AND MODULES** **Functions:** Decomposition of a program into independent, reusable sub-problems; parameters, return values and scope. **Lambda Expressions and Functional Tools:** lambda, map, filter, and reduce, concise transformations and their use cases. **Recursion:** Algorithmic thinking applied to self-similar sub-problems. Algorithmic Efficiency: Informal comparison of iterative vs. recursive approaches and their time trade-offs. **Modules and Libraries:** Importing and creating modules, packages and built-in libraries; responsible AI-assisted coding practices.	9
UNIT – 4	**OBJECT ORIENTED PROGRAMMING** **Introduction to OOP**: Features of object-oriented programming, encapsulation, abstraction, inheritance, polymorphism. **Classes and Objects:** Class definition, the __init__ method, class variables and instance variables, public and private data members, calling methods from other methods, built-in class attributes. **Inheritance:** Inheriting classes, types of inheritance, method overriding. **Polymorphism and Operator Overloading:** Polymorphism through method overriding; operator overloading. **Composition:** Composition (has-a) vs. inheritance (is-a) relationships.	9
UNIT – 5	**FILE HANDLING, EXCEPTIONS AND DATA ANALYSIS** **File Handling:** Reading and writing files; decomposition of a data workflow into read/process/write stages. **Exception Handling**: try-except-finally; abstraction of error cases into predictable, manageable patterns; user-defined exceptions. **Data Analysis and Visualization**: Introduction to NumPy, pandas and Matplotlib basics; pattern recognition in real datasets.	9
	Total	**45**

Text Books:

1. Downey, A. B. (2015). Think Python: How to think like a computer scientist (2nd ed.). O'Reilly Media.
2. Guttag, J. V. (2021). Introduction to computation and programming using Python: With application to computational modeling and understanding data (3rd ed.). MIT Press.

Reference Books:

1. Sweigart, A. (2025). Automate the boring stuff with Python: Practical programming for total beginners (3rd ed.). No Starch Press.
2. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.
3. Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215
4. Jose, J. (2019). Taming Python by programming (2nd ed.). Khanna Book Publishing.

Web References:

1. Python Software Foundation. (n.d.). The Python tutorial. https://docs.python.org/3/tutorial/
2. W3Schools. (n.d.). Python tutorial. https://www.w3schools.com/python/
3. NPTEL. (n.d.). Programming, data structures and algorithms using Python. https://nptel.ac.in
4. Real Python. (n.d.). Python tutorials. https://realpython.com/

COURSE CODE	SUSTAINABILITY AND GREEN ENGINEERING	Category ESC	L-T-P-C 2-0-0-2

Pre-requisite: Basic knowledge of Engineering Sciences and Environmental Studies

Course Objectives: The course is designed to:

- Understand the fundamental concepts of sustainability, sustainable development, environmental systems and the role of engineering in addressing global and local sustainability challenges.
- Explain the principles of green engineering, resource efficiency, renewable energy, pollution prevention, cleaner production and sustainable materials.
- Apply life-cycle thinking, life-cycle assessment, carbon footprinting and circular economy principles for evaluating engineering products, processes and systems.
- Develop the ability to propose technically feasible and environmentally responsible engineering solutions considering environmental, economic and social dimensions of sustainability.

Course Outcomes: After successful completion of the course, the students will be able to

		Knowledge Level (K)#
CO1	Explain the fundamental concepts of sustainability, sustainable development, SDGs, environmental limits, resource depletion and the role of engineering in sustainable development	K2
CO2	Describe the principles of green engineering, cleaner production, renewable energy, sustainable materials, energy efficiency and pollution prevention	K2
CO3	Apply life-cycle thinking, life-cycle assessment concepts, carbon-footprint estimation, material-flow analysis and circular-economy principles to engineering products and processes	K3
CO4	Analyze engineering systems with respect to energy, material, water and waste efficiency and identify opportunities for improving their environmental performance	K3
CO5	Evaluate alternative engineering solutions and recommend sustainable and green-engineering interventions considering technical, environmental, economic and social criteria	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	L	L	L	–	–	L	H	L	L	L	–			
CO2	H	L	L	L	L	L	H	L	L	L	L			
CO3	L	H	L	H	H	L	H	L	L	L	L			
CO4	L	H	H	L	H	L	H	L	L	L	L			

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO5	L	H	H	L	H	H	H	H	L	L	H			

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	Fundamentals of Sustainability and Sustainable Development: Introduction to Sustainability; Meaning and dimensions of sustainability; Sustainability as an engineering challenge; Environmental, economic and social dimensions; Sustainability principles and systems thinking; Interdependence of engineering, society and environment; Sustainable development – concept and evolution; Brundtland definition and principles of sustainable development; Sustainable Development Goals (SDGs); SDGs and engineering applications; Population growth, urbanization and resource consumption; Planetary boundaries and ecological limits; Natural resource depletion; Ecological footprint and resource footprint; Climate change and global environmental challenges; Role and responsibilities of engineers in sustainable development.	6
UNIT – 2	Principles of Green Engineering and Cleaner Production: Concept and need for green engineering; principles of green engineering; prevention and resource efficiency; sustainable design and Design for Environment (DfE); design for disassembly and recyclability; cleaner production; pollution prevention; waste minimization; process modification and material substitution; energy and water conservation; green manufacturing; industrial symbiosis; green supply chains.	6
UNIT – 3	Life-Cycle Thinking, LCA and Carbon Management: Life-cycle thinking; product life cycle and system boundaries; cradle-to-grave and cradle-to-cradle concepts; Life-Cycle Assessment (LCA) – concept, objectives and applications; four phases of LCA – goal and scope, LCI, LCIA and interpretation; functional unit; comparative assessment of engineering alternatives; greenhouse gases; carbon footprint; Scope 1, 2 and 3 emissions; carbon reduction strategies; water and ecological footprints.	6
UNIT – 4	Sustainable Energy, Materials, Buildings and Circular Economy: Sustainable Energy: Energy efficiency and conservation; renewable energy – solar, wind and biomass; energy storage; smart energy systems; net-zero energy. Sustainable Materials: Sustainable material selection; embodied energy; recycled, recyclable and bio-based materials; sustainable construction materials; durability and life extension. Green Buildings: Principles of green buildings; energy- and water-efficient buildings; passive design; sustainable materials; indoor environmental quality; LEED and GRIHA; net-zero buildings.	6

	Circular Economy: Linear versus circular economy; 3R/5R principles; reuse, repair, refurbishment and recycling; industrial symbiosis; EPR; waste-to-resource approaches; circular business models.	
UNIT – 5	Sustainable Engineering Applications and Design: Water conservation and reuse; rainwater harvesting; wastewater and solid waste management; e-waste and waste-to-energy; sustainable mobility and electric vehicles; public and non-motorized transportation; transportation energy efficiency and life-cycle impacts; IoT and AI for resource optimization; smart grids and buildings; green computing; sustainability indicators; multi-criteria decision making; sustainable design and technical, economic and environmental trade-offs; risk, resilience and ethics; Indian case studies; mini design problem.	6
	Total	**30**

Text Books:

1. Santosh Kumar Garg and Ranjini Garg, Environmental Studies and Green Technologies; Khanna Publishers, New Delhi.
2. Sanjay Kumar Samantarai, Biswajit Rath, Jogeswari Rout, Smaranika Das, Diptikanta Acharya and Shantanu Kumar Sahu; Green Technology: An Innovative Solution for Environmental Sustainability, Newredmars Education, 2025.

Reference Books:

1. David T. Allen and David R. Shonnard, Sustainable Engineering: Concepts, Design and Case Studies, Pearson.
2. David T. Allen and David R. Shonnard, Green Engineering: Environmentally Conscious Design of Chemical Processes, Prentice Hall.
3. Ni-Bin Chang, Systems Analysis for Sustainable Engineering: Theory and Applications, McGraw-Hill.

Web References:

1. NPTEL – Sustainable Engineering Concepts and Life Cycle Analysis
2. NPTEL – Business and Sustainable Development
3. NPTEL – Environment and Development
4. NPTEL – Sustainable Transportation Systems
5. NPTEL – Energy and Sustainability

COURSE CODE	DATA STRUCTURES USING C LAB	Category ESC	L-T-P-C 0-0-3-1.5

Pre-requisite: Programming for Problem Solving using C

Course Objectives: The main objectives of the course are to

- Build hands-on programming proficiency in C and to train students to translate the design of a data structure into a correct, efficient working program.
- Strengthen the ability to trace, debug and test code, and to choose an appropriate data structure and implementation strategy for a given problem

Course Outcomes: *At the end of the course, student will be able to*

CO	Course Objective Description	Knowledge Level (K)#
CO1	Write and debug C programs using pointers and dynamic memory allocation to implement array-based searching and sorting techniques.	K3
CO2	Build and manipulate singly, doubly and circular linked lists, and use them to solve problems such as polynomial arithmetic and duplicate removal.	K3
CO3	Implement stack-based and queue-based programs for expression conversion/evaluation, parenthesis matching and recursion-related problems.	K3
CO4	Construct binary search trees and implement all traversal techniques to solve hierarchical-data problems.	K3
CO5	Design and test hash-table programs using chaining and open-addressing collision-resolution techniques.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	H	M	-	H	-	-	L	L	-	-	-	M	-
CO2	H	M	H	-	H	-	-	L	L	-	-	-	M	-
CO3	H	H	M	-	H	-	-	-	L	-	-	-	M	-
CO4	H	H	H	M	H	-	-	L	L	-	L	L	H	L
CO5	H	H	M	-	H	-	-	L	L	-	-	M	H	L

Lab Requirements (tools / software to carry out the experiments):

- C compiler: GCC (MinGW-w64 on Windows / built-in on Linux) with an IDE such as Code::Blocks,
- VS Code or Dev-C++.
- Debugger: GDB, to trace pointers, linked lists and recursion step-by-step.
- Memory checker: Valgrind (on Linux) to detect memory leaks and invalid access in dynamic-memory programs.
- Visualization aids: VisuAlgo, USF Data-Structure Visualizations, or the IIIT-H Virtual Labs for lists,
- trees and hashing.
- Operating system: Windows or Linux, 64-bit, with 4 GB RAM or more.
- No-install alternative: OnlineGDB or Replit for quick compilation and sharing.

List of Experiments

Exercise 1: Array-Based Techniques, searching

1. Write a program using pointers to reverse an array in place, without using a second array.
2. Write a program to dynamically allocate an array using malloc, resize it using realloc, and free it correctly.
3. Implement linear search on an unsorted array; count and print the number of comparisons made.
4. Implement binary search on a sorted array; count the comparisons made and compare the count with linear search on the same data.

Exercise 2: Sorting

1. Implement bubble, selection and insertion sort; print the array after each pass, and tabulate the comparisons and swaps each algorithm performs on the same input.
2. Implement Quick sort and Merge sort; print the array after every partition/merge step to observe how each algorithm progresses.

Exercise 3: Singly Linked List

1. Implement a singly linked list with creation, traversal, insertion and deletion of nodes.
2. Implement a singly linked list with insertion at the beginning, end and a given position.
3. Implement deletion of a node by value and by position with proper handling of an empty list, then reverse the list iteratively and recursively.

Exercise 4: Singly Linked List Applications

1. Detect and remove duplicate nodes from an unsorted linked list.
2. Find the middle node of a singly linked list in a single traversal.
3. Merge two sorted singly linked lists into a single sorted list.

Exercise 5: Doubly and Circular Linked Lists

1. Implement a doubly linked list with insertion, deletion and forward/backward traversal.
2. Implement a circular linked list and demonstrate insertion, deletion and traversal without infinite looping.
3. Represent two polynomials using linked lists and write a program to add them.

Exercise 6: Stack Implementation

1. Implement a stack using an array and using a linked list, with overflow/underflow checks.
2. Write a program to check whether an expression has balanced parentheses using a stack.

Exercise 7: Stack Applications

1. Write a program to convert an infix expression to postfix using a stack.
2. Write a program to evaluate a postfix expression using a stack.
3. Write a program to check whether a given string is a palindrome using a stack.

Exercise 8: Recursion and Stack Simulation

1. Write recursive programs for factorial, Fibonacci and sum of digits.
2. Simulate the run-time call stack of a recursive function by writing an equivalent iterative version using an explicit stack.

Exercise 9: Queue and Circular Queue

1. Implement a circular queue and demonstrate that it avoids the false-overflow problem of a linear queue.
2. Implement a queue using an array and using a linked list.

Exercise 10: Binary Trees

1. Implement a Binary Tree with insertion and search operations.
2. Implement deletion of a node from a binary tree, handling all three cases (leaf, one child, two children).

Exercise 11: Binary Search Trees

1. Implement a Binary Search Tree (BST) with insertion and search operations.
2. Implement deletion of a node from a BST, handling all three cases (leaf, one child, two children).

Exercise 12: Tree Traversals and Problems

1. Implement a Binary Search Tree (BST) with insertion, and perform in-order, pre-order and post-order traversals.
2. Write a program to find the height of a binary tree and count the total number of nodes and leaf nodes.
3. Write a program to check whether a given binary tree is a valid Binary Search Tree.

Exercise 13: Hashing

1. Implement a hash table using chaining for collision resolution, with insertion, search and deletion.
2. Implement a hash table using linear probing.

Web References:

1. **NPTEL – Programming, Data Structures and Algorithms using C (IIT Madras)**
   https://www.nptel.ac.in/courses/106106127
2. **NPTEL – Data Structures and Algorithms (IIT Delhi)**
   https://www.nptel.ac.in/courses/106102064
3. **NPTEL – Programming, Data Structures and Algorithms using C – Course Archive**
   https://onlinecourses-archive.nptel.ac.in/noc18_cs25/preview
4. **Virtual Labs – Data Structures I (IIIT Hyderabad)**
   https://ds1-iiith.vlabs.ac.in/
5. **Virtual Labs – IIIT Hyderabad Data Structures Labs**
   https://www.vlab.co.in/participating-institute-iiit-hyderabad
6. **GeeksforGeeks – Data Structures and Algorithms**
   https://www.geeksforgeeks.org/dsa/

COURSE CODE	DESIGN THINKING AND INNOVATION	Category ESC	L-T-P-C 0-1-3-2.5

Pre-requisites: NIL

Course Objectives: At the end of the course, the student should be able to:

- Introduce students to the principles, mindset and methodologies of **Design Thinking and human-centered problem solving**.
- Develop students' ability to **observe, empathize, analyze user needs and formulate well-defined engineering and societal problems**.
- Develop students' **creative and innovative thinking abilities** through systematic ideation, concept generation and evaluation techniques.
- Enable students to **develop, prototype, test and iteratively improve solutions** using user feedback and appropriate physical/digital tools.
- Develop students' ability to **identify innovation opportunities, create value propositions, understand basic IPR and communicate innovative solutions effectively**.

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

1. Explain the principles, processes and mindset of Design Thinking for engineering problem-solving and innovation (K2)
2. Apply empathy, observation and problem-framing techniques to identify user needs and define engineering problems (K3)
3. Apply creative and systematic ideation techniques to generate and select innovative solution concepts (K3)
4. Analyze solution concepts and prototypes using user feedback and appropriate design criteria for iterative improvement (K4)
5. Analyze the innovation potential of solutions considering value proposition, feasibility, sustainability, societal impact and basic IPR aspects (K4)

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	M	M	L	–	–	L	–	–	L	L	M			
CO2	M	H	H	L	L	H	M	L	M	M	M			
CO3	L	M	H	L	M	L	L	–	H	M	M			
CO4	H	H	H	M	H	M	M	L	H	M	M			
CO5	L	M	H	L	M	H	H	M	M	H	H			

UNIT	CONTENTS	Contact Hours
UNIT - 1	Design Thinking Foundations & Empathy Introduction to Design Thinking ; Design thinking mindset and principles ; Design thinking vs. conventional engineering problem-solving ; Human-centered design ; Five-stage design thinking process ; Creativity, curiosity and observation ; Understanding users and stakeholders ; Empathy and user research ; Interviews, observation and contextual inquiry ; Empathy mapping ; Personas and user journeys ; Identifying explicit and latent/unarticulated needs ; Ethical considerations in user research Practical Activities Observation of a real-life problem Conducting user interviews Preparing	

UNIT	CONTENTS	Contact Hours
	an empathy map Developing a persona Preparing a user journey map Unit Outcome Students will be able to understand users, identify their needs and translate observations into meaningful design opportunities.	
UNIT - 2	Problem Definition, Analysis & Opportunity Identification From observations to insights ; Data synthesis and affinity mapping ; Identifying patterns and pain points ; Problem statements ; Point-of-View (POV) statements ; "How Might We" (HMW) questions ; Stakeholder mapping ; 5 Whys and root-cause analysis ; Problem prioritization ; Need–Opportunity mapping ; Problem validation ; Design constraints and specifications ; Social, environmental and sustainability considerations ; Identifying opportunities for innovation Practical Activities Affinity mapping of interview data Develop three alternative problem statements Prepare POV and HMW statements Stakeholder map Root-cause analysis Select and validate one problem/opportunity Unit Outcome Students will be able to analyze user research, identify root causes, frame well-defined problems and recognize opportunities for innovative solutions.	
UNIT - 3	Creativity, Ideation & Concept Development Creativity and innovation mindset; Divergent and convergent thinking ; Brainstorming and brainwriting ; SCAMPER ; Mind mapping ; Reverse brainstorming ; Analogical thinking ; Biomimicry ; Lateral thinking ; Forced connections ; Morphological analysis ; TRIZ — basic principles ; Individual and collaborative ideation ; AI-assisted ideation: opportunities and limitations ; Idea clustering and screening ; Impact–feasibility matrix ; Concept selection ; Design criteria and decision matrix Practical Activities Generate 20–30 ideas for a selected problem SCAMPER exercise Brainwriting exercise Biomimicry-based ideation AI-assisted idea generation and critical evaluation Prepare an idea-selection matrix Select the top 2–3 concepts Unit Outcome Students will be able to generate diverse solution concepts using creative techniques and systematically select promising ideas based on defined criteria.	
UNIT - 4	Prototyping, Experimentation & Validation Concept development ; Prototype mindset ; Types of prototypes: Paper prototype, Physical prototype, Digital prototype, Storyboard, Role-play, Service prototype ; Low-fidelity and high-fidelity prototypes ; Rapid prototyping ; "Fail fast, learn fast" ; Minimum Viable Prototype (MVP) ; Design experimentation ; Prototype evaluation ; User testing ; Think-aloud method ; Feedback collection ; Usability and functionality testing; Iterative design ; Design critique ; Documentation of iterations ; Use of CAD, simulation and AI-assisted visualization tools Practical Activities Develop a low-fidelity prototype Prepare a storyboard Build a physical/digital prototype Conduct user testing Collect structured feedback Modify the prototype Prepare an iteration log	
UNIT - 5	Innovation, Value Creation & Communication Innovation: meaning, characteristics and importance: Invention vs. innovation ; Incremental, disruptive and radical innovation ; Product, process, service and social innovation ; Technology-driven vs. need-driven innovation ; Sustainable and frugal innovation ; Innovation opportunities and emerging technologies ; From prototype to innovation ; Desirability, feasibility and viability ; Value Proposition ; Value Proposition Canvas ; Introduction to Business Model Canvas ; Customer segments and stakeholder value; Innovation validation; Market/user validation ; Intellectual Property Rights — introduction; Patents, copyright, trademarks and design	

UNIT	CONTENTS	Contact Hours
	protection ; Responsible innovation ; Inclusive and sustainable innovation ; Innovation ecosystem and entrepreneurship ; Communicating innovative ideas ; Innovation pitch Practical Activities Classify existing products/services as incremental, disruptive or radical innovations. Identify three innovation opportunities from a campus/community problem. Prepare a Value Proposition Canvas. Validate the proposed innovation with potential users. Prepare an introductory Business Model Canvas. Identify possible IPR protection. Prepare and deliver a 3-minute Innovation Pitch.	

Learning Resources:

Textbook:

1. Chandan Adhikari, Prabir Kumar Das & Tanay Pramanik, Textbook of Design Thinking and Innovation, S. Chand Publishing, 2026.
2. Shalini Rahul Tiwari & Rohit Rajendra Swarup, Design Thinking: A Comprehensive Textbook, Wiley India, 2024

Reference Books:

1. Soni, Pavan, Design Your Thinking: The Mindsets, Toolsets and Skill Sets for Creative Problem-Solving, Penguin Random House India.
2. Jauhari, Vinnie & Sudhanshu Bhushan, Innovation Management, Oxford University Press India.

Web Resources:

1. Atal Innovation Mission, NITI Aayog – Innovation and Design Thinking Resources. https://aim.gov.in/
2. Institution's Innovation Council (IIC), Ministry of Education, Government of India, https://iic.mic.gov.in/
3. Startup India, Government of India – Innovation and Entrepreneurship Ecosystem, https://www.startupindia.gov.in/
4. National Innovation Foundation – India – Grassroots and Frugal Innovation, https://nif.org.in/
5. Stanford d.school – Design Thinking Bootleg and Design Thinking Resources, https://dschool.stanford.edu/resources/design-thinking-bootleg
6. IDEO.org – Human-Centered Design and Social Innovation Resources, https://www.ideo.org/
7. IDEO U – Design Thinking and Innovation Resources, https://www.ideou.com/pages/design-thinking
8. NPTEL – Online Courses on Design, Innovation, Creativity and Entrepreneurship, https://nptel.ac.in/
9. SWAYAM – Online Courses on Innovation, Entrepreneurship and Design Thinking, https://swayam.gov.in/

COURSE CODE	AI TOOLS AND APPLICATIONS	Category ESC	L-T-P-C 0-1-3-2.5

Pre-requisites: NIL

Course Objectives: To familiarize students with the landscape of AI tools and their underlying working principles.

1. To develop competence in prompt engineering for effective interaction with generative AI systems.
2. To enable students to apply AI tools for data analysis, visualization, and interpretation.
3. To train students in using AI-assisted tools for coding, writing, and productivity tasks.
4. To instill awareness of ethical, legal, and societal considerations in AI tool usage.

Course Outcomes: *After successful completion of this course, students will be able to:*

1. Recall the categories, capabilities, and limitations of major AI tools (text, image, code, data). (K1)
2. Explain the principles of prompt engineering and how generative AI models process instructions. (K2)
3. Apply AI tools to solve first-year engineering problems in analysis, writing, and coding tasks. (K3)
4. Analyze AI-generated outputs for accuracy, bias, and reliability before use. (K4)
5. Evaluate the ethical, academic-integrity, and societal implications of AI tool use, justifying appropriate usage choices. (K5)

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	M	–	–	–	H	L	L	–	–	–	L			
CO2	M	M	–	–	H	L	–	L	–	M	L			
CO3	H	H	M	M	H	L	L	–	L	M	M			
CO4	M	H	M	H	H	M	M	H	L	M	M			
CO5	L	M	M	M	M	H	H	H	M	H	M			

Unit 1: **Introduction to AI and the AI Tools Landscape**

What is AI, Machine Learning, and Generative AI — basic definitions and how they relate to one another.

Core categories of AI tools: conversational/text, code assistants, data-analysis tools, image-generation tools, and writing/productivity tools.

How large language models process and generate responses (conceptual, non-mathematical overview).

Criteria for selecting the right AI tool for a given task — accuracy needs, cost, data sensitivity, and output format.

Setting up accounts and workspaces; understanding free-tier versus paid-tier capabilities and limitations.

Widely Used AI Tools by Application (for classroom coverage):

Category	Example Tools	Typical Application
Conversational / General-Purpose AI	ChatGPT, Claude, Gemini, Perplexity	Writing, research, coding help, file/data analysis, cited web search

AI Coding Assistants	GitHub Copilot, Cursor, Claude Code	In-IDE code completion, AI-native editing, multi-file agentic coding
Data Analysis Tools	ChatGPT (Advanced Data Analysis), Microsoft Copilot in Excel, Julius AI	Upload a dataset, generate statistics/charts, spreadsheet formulas via natural language
Image Generation	Midjourney, DALL·E 3, Stable Diffusion / Flux	Illustrations, concept art, design mockups
Writing & Productivity	Grammarly, Gamma, Fathom, Notion AI	Grammar/style checking, slide generation, meeting summaries, task automation
Audio, Video & Presentation (optional)	ElevenLabs, Runway/Sora, Synthesia	Text-to-speech, AI video generation, script-to-video presentations

Teacher Practical Examples (for classroom demonstration):

1. Live comparison of the same general engineering query (e.g., "explain the engineering design process") answered by two different AI chatbots, highlighting differences in response style and accuracy.
2. Walkthrough of an AI tool's interface — settings, conversation history, file upload, and model selection — using a common platform.

Student Exercise Problems (for practice):

1. Create accounts on two AI platforms and compare their responses to a simple engineering question (e.g., "explain the difference between analysis and design").
2. Prepare a one-page comparison chart of five AI tools by category and primary use case.
3. Write a short reflection identifying one task in your daily academic life where an AI tool could help.

Unit 2: Prompt Engineering and Generative AI Tools

Anatomy of a good prompt — context, clear instruction, desired format, and constraints.

Zero-shot, few-shot, and chain-of-thought prompting techniques and when to use each.

Role-based and context-setting prompts (e.g., asking the AI to respond as a reviewer, tutor, or domain expert).

Iterative prompt refinement — diagnosing a poor AI response and rewriting the prompt to fix it.

Using generative AI for brainstorming, summarizing, and drafting technical content.

Teacher Practical Examples (for classroom demonstration):

1. Refine a vague prompt (e.g., "explain project scheduling") into a precise one step-by-step, showing output improvement at each stage.
2. Demonstrate few-shot prompting by providing examples to guide the AI toward a specific output format (e.g., a formatted comparison table).

Student Exercise Problems (for practice):

1. Write three versions of a prompt (basic, detailed, role-based) for the same task and compare outputs.
2. Use an AI tool to summarize a one-page technical article and verify the summary against the original.
3. Draft a chain-of-thought prompt to solve a multi-step numerical/logical problem (e.g., a unit-conversion or estimation problem) and check the reasoning.

Unit 3: AI Tools for Data Analysis and Visualization

AI-assisted spreadsheet and dataset analysis using natural-language commands.

Natural-language querying of datasets to extract summary statistics and insights.

AI tools for generating charts, graphs, and visual summaries from tabular data.

Interpreting AI-generated statistical summaries correctly and recognizing their limitations.

Identifying and correcting errors or misinterpretations in AI-generated data analysis.

Teacher Practical Examples (for classroom demonstration):

1. Upload a small dataset (e.g., class attendance or survey data) to an AI tool and generate a summary-statistics report and chart using natural-language commands.
2. Show an example where an AI tool misinterprets data (e.g., treats a numeric code as text), and demonstrate how to verify and correct it.

Student Exercise Problems (for practice):

1. Use an AI tool to clean and summarize a small sample dataset (e.g., class attendance or marks data).
2. Generate three different chart types from the same dataset using AI prompts and justify which is most appropriate.
3. Cross-check one AI-generated statistical claim manually and report any discrepancy found.

Unit 4: AI Tools for Coding, Writing, and Productivity

AI coding assistants — code generation, explanation, and debugging support for simple programs.

AI writing assistants for reports, emails, and technical documentation.

AI tools for scheduling, note-taking, and meeting/lecture summarization.

AI-assisted presentation and slide-outline generation from a topic prompt.

Integrating AI tools into an efficient day-to-day academic and engineering workflow.

Teacher Practical Examples (for classroom demonstration):

1. Use an AI coding assistant to generate a simple program (e.g., a temperature converter), then intentionally introduce a bug and have the AI debug it.
2. Generate a presentation outline and speaker notes from a topic prompt using an AI writing/productivity tool.

Student Exercise Problems (for practice):

1. Use an AI coding tool to write and debug a simple program for a given problem statement.
2. Draft a formal email or short report using an AI writing assistant, then manually edit it for accuracy and tone.
3. Generate a slide outline for a 5-minute seminar topic using an AI tool and refine it.

Unit 5: Ethics, Reliability, and Emerging Trends

Bias and hallucination in AI outputs, and techniques for detecting and fact-checking them.

Academic integrity and appropriate citation of AI assistance in coursework and submissions.

Data privacy and security considerations, including obligations under India's Digital Personal Data Protection Act, 2023 (DPDP Act), when uploading personal or institutional data.

Copyright and intellectual property considerations in AI-generated text and image content.

Emerging trends — multimodal AI and AI agents — and their implications for future engineering practice.

Teacher Practical Examples (for classroom demonstration):

1. Show a documented case of AI hallucination (a fabricated fact or citation) and demonstrate how to fact-check it.
2. Walk through an example of properly citing AI assistance in an academic submission per institutional guidelines.
3. Walk through a worked example of checking whether AI-generated text or image output raises a copyright/IP concern (e.g., close paraphrasing of a source).

Student Exercise Problems (for practice):

1. Identify and document a hallucination or bias instance produced by an AI tool on a factual query.
2. Draft an "AI usage declaration" for a hypothetical assignment, citing tools and extent of use appropriately.
3. Review the data-handling/privacy policy of one AI tool used in this course and identify one DPDP Act, 2023 compliance consideration relevant to student data.

Learning Resources:

Text Books

1. Marr, B. (2024). Generative AI in practice: 100+ amazing ways generative artificial intelligence is changing business and society. Wiley.
2. Mollick, E. (2024). Co-intelligence: Living and working with AI. Portfolio/Penguin.

Reference Books

1. Selwyn, N. (2022). Education and technology: Key issues and debates (3rd ed.). Bloomsbury.
2. Susskind, R., & Susskind, D. (2022). The future of the professions: How technology will transform the work of human experts (Updated ed.). Oxford University Press.
3. Russell, S., & Norvig, P. (2021). Artificial intelligence: A modern approach (4th ed.). Pearson.

Web Resources

1. Google. (n.d.). Google AI essentials. Grow with Google. https://grow.google/ai-essentials/
2. OpenAI. (n.d.). OpenAI documentation and guides. https://platform.openai.com/docs
3. Anthropic. (n.d.). Prompt engineering overview. Claude Docs. https://docs.claude.com/en/docs/build-with-claude/prompt-engineering/overview
4. UNESCO. (2023). Guidance for generative AI in education and research. https://unesdoc.unesco.org/
5. AICTE. (2025). AICTE inputs on Artificial Intelligence — Year of AI initiative. https://www.aicte.gov.in/downloads/initiatives/AICTE%20inputs%20on%20Artificial%20Intelligence%20-compre hensive.pdf
6. Ministry of Electronics and Information Technology, Government of India. (2023). Digital Personal Data Protection Act, 2023. https://www.meity.gov.in/content/digital-personal-data-protection-act-2023

COURSE CODE	PHYSICS FOR ADVANCED COMPUTING LAB	Category BSC	L-T-P-C 0-0-3-1.5

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1														
CO2														
CO3														
CO4														
CO5														

LIST OF EXPERIMENTS

(Any **TEN** of the following listed experiments)
(Out of which any **TWO** experiments may be conducted in virtual mode)

1. Determination of the thickness of a thin object by the wedge method.
2. Determination of the radius of curvature of a given plano convex lens by forming Newton's rings.
3. Determination of wavelengths of different spectral lines in the mercury spectrum using a diffraction grating in normal incidence configuration.
4. Determination of the resolving power of a telescope.
5. Determination of the dielectric constant of a given capacitor by the charging and discharging method.
6. Study the variation of B versus H by magnetizing the magnetic material (B-H curve).
7. Determination of temperature coefficients of a thermistor.
8. Determination of the wavelength of a Laser and a grating element by using a grating.
9. Estimation of Planck's constant using the photoelectric effect.
10. Determination of the Resistivity of a semiconductor by the four probe method.
11. Determination of the energy gap of a semiconductor using a p-n junction diode.
12. Study the variation of magnetic field along the axis of a current carrying circular coil by Stewart & Gee's method.
13. Determination of Hall voltage and Hall coefficient of a given semiconductor using Hall Effect.
14. Determination of Acceleration due to gravity and Radius of gyration using a Compound pendulum.
15. Determination of magnetic susceptibility by Kundt's tube method.
16. Verification of Brewster's law.
17. Sonometer: Verification of laws of stretched string.
18. Determination of frequency of electrically maintained tuning fork by Melde's apparatus.

References:

1. Balasubramanian, S., & Srinivasan, M. N. (2017). A text book of practical physics. S. Chand Publishers.

V-Lab: vlab.co.in

COURSE CODE	PYTHON PROGRAMMING LAB	Category ESC	L-T-P-C 0-0-2-1.0

Pre-requisite: Concurrent with Python Programming

Course Objectives: The main objectives of the course are to

- Develop proficiency in Python control structures while explicitly linking conditionals to abstraction and loops to pattern recognition.
- Build competence in selecting Python data structures (strings, lists, tuples, dictionaries)
- Introduce object-oriented programming concepts such as classes, objects, inheritance, and polymorphism, enabling students to model real-world problems through encapsulation and abstraction.
- Train students to decompose problems into functions and recursive sub-problems, use lambda expressions and functional tools, reason informally about algorithmic efficiency, and use AI coding assistants responsibly.
- Equip students with file handling, exception handling, and basic data analysis/visualisation skills, framed as decomposing a larger data workflow into manageable, robust steps and to integrate these skills in a capstone mini-project.

Course Outcomes : *At the end of the course, student will be able to*

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Develop Python programs using variables, data types, operators, input/output, conditional statements, loops, and appropriate coding practices to solve computational problems.	K3
CO2	Implement Python programs using strings, lists, tuples, dictionaries, and comprehensions to efficiently represent, manipulate, and process different types of data.	K3
CO3	Develop and analyze modular Python programs using functions, lambda expressions, functional programming tools, recursion, modules, and libraries, considering basic algorithmic efficiency.	K4
CO4	Design and implement object-oriented Python programs using classes, objects, encapsulation, inheritance, polymorphism, operator overloading, and composition to model real-world problems.	K4
CO5	Develop Python applications that perform file processing and exception handling, and apply NumPy, pandas, and Matplotlib for basic data analysis and visualization.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	H	M	M	-	H	-	-	L	L	-	M	H	M	L
CO2	H	H	H	M	H	-	-	L	L	-	M	H	H	L
CO3	H	H	H	M	H	-	-	M	L	-	H	H	H	M
CO4	H	H	H	M	H	-	-	M	M	L	H	H	M	M
CO5	H	H	H	H	H	M	-	M	M	L	H	H	H	M

Lab Requirements (tools / software to carry out the experiments):

- Python 3.x interpreter (CPython, from python.org).
- IDE / editor: Thonny or IDLE (beginner-friendly), VS Code (Python extension) or PyCharm Community.
- pip package manager with the NumPy, pandas and Matplotlib libraries for the data-analysis exercises.
- Notebook environment: Jupyter Notebook or Google Colab (no install) for experimentation and plotting.
- Operating system: Windows, Linux or macOS, 64-bit, with 4 GB RAM or more.
- No-install alternative: Google Colab, Replit or Programiz Python for practice in a browser.

List of Experiments

Exercise 1: Getting Started with Python

1. Write and run a Python program that prints "Hello, World!" along with the Python version in use. Note the difference between running a saved script from the terminal and typing the same statements in the interactive shell.
2. Write a Python program that reads a name and an age from the user and prints a personalised greeting.
3. Write a Python program that reads two numbers and prints their sum, difference, product, quotient and remainder, each on a labelled line.
4. Write a Python program to compute simple interest given principal, rate, and time.
5. Write a Python program to compute the area and perimeter of a rectangle from user input.
6. Write a Python program to swap two numbers without using a third variable.

Exercise 2: Conditionals

1. Write a Python program to compute the Body Mass Index (BMI) from height and weight, rounded to two decimal places, and classify it as Underweight, Normal, Overweight or Obese using if-elif-else.
2. Write a menu-driven Python program to convert a temperature between Celsius, Fahrenheit and Kelvin based on the user's choice.
3. Write a Python program to find the largest of three numbers.
4. Write a Python program to Check whether a given year is a leap year.

Exercise 3: Write a Python program to

1. Determine whether a number is positive, negative, or zero.
2. Check if a given number is prime
3. Convert temperature between Celsius and Fahrenheit based on user choice
4. Compute and display the area of a valid triangle using Heron's formula, with inline comments explaining each step.

Exercise 4: Loops, Number Problems and Patterns

1. Use a for loop to print the first n Fibonacci terms, and a while loop to test whether a number is prime (using break, printing the number of division tests).
2. Use a while loop to compute the GCD and LCM of two integers using Euclid's algorithm.
3. Find the first n prime numbers using nested loops with break.
4. Sum of first N natural numbers

Exercise 5: String Processing

1. Write a function that checks whether a string is a palindrome using slicing and string methods, ignoring spaces, punctuation and case; justify whether a string or a list of characters is the better structure.
2. Write a script that counts the vowels, consonants, digits and special characters in a user-provided paragraph.
3. Write a script that sanitizes a full name with irregular capitalization and excess spaces using .strip() and .title(), and extracts the initials.

Exercise 6: Lists and List Comprehensions

1. Remove duplicates from a list of sensor readings, first without set() and then with set(); compare the two approaches for time complexity and readability.
2. Use a list comprehension to filter out noise readings outside a realistic range (e.g., -40°C to 85°C).
3. Compute and display the average, maximum and minimum of the filtered list.

Exercise 7: Write a Python Program to

1. Find the second-largest element in a list of exam scores.
2. Merge two sorted lists of student roll numbers into a single sorted list.
3. Generate squares of all even numbers from 1 to 50 using list comprehension.

Exercise 8: Dictionaries and Tuples

1. Create a dictionary of student records keyed by roll number with (Name, Marks) tuples as values; justify why a dictionary is preferable to a list of tuples for lookup.
2. Build a word-frequency dictionary from a text paragraph (case-insensitive). (Count the frequency of each word in a paragraph using a dictionary)
3. Search the student dictionary by roll number, assign grades and identify the highest scorer.

Exercise 9: Recursion vs Iteration

1. Write a recursive function fibonacci_recursive(n) that returns the n-th Fibonacci term.
2. Write an iterative function fibonacci_iterative(n) using a loop.
3. Compute the GCD of two numbers recursively using the Euclidean algorithm.
4. Compute factorial using recursion

Exercise 10: File Handling and Modular Workflow

1. Create a file student_records.txt and write five lines of data formatted as RollNo, Name, Marks.
2. Read the file, identify students scoring above 75 and write them to distinction_list.csv using context managers.
3. Count the words, lines, and characters in a given text file.
4. Read a text file of daily temperature readings and write the maximum, minimum, and average to an output file

Exercise 11: Exception Handling

1. Write a program to compute the real roots of a quadratic equation, handling all cases (real distinct, real equal, no real roots).
2. Wrap the parsing and calculation in a try-except block handling ValueError and ZeroDivisionError, with a finally block logging attempts to calculation_log.txt.
3. Package the logic into a reusable function solve_quadratic(a, b, c) suitable for import by other modules.

Reference Books:

1. Sweigart, A. (2025). Automate the boring stuff with Python: Practical programming for total beginners (3rd ed.). No Starch Press.

2. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.
3. Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Web References:

- Python Software Foundation – The Python Tutorial: https://docs.python.org/3/tutorial/
- NPTEL – Python for Data Science: https://nptel.ac.in

COURSE CODE	INDIAN KNOWLEDGE SYSTEMS	Category MNC	L-T-P-C 2-0-0-0

Pre-requisites: NIL

Course Objectives: *The course aims to*:s

- Introduce students to the philosophical foundations, historical evolution, and fundamental principles of Indian Knowledge Systems.
- Familiarize students with the multidisciplinary contributions of IKS in areas such as mathematics, astronomy, medicine, ecology, governance, arts, and technology.
- Develop students' ability to analyze and evaluate the contemporary relevance of traditional Indian knowledge in addressing challenges related to sustainability, health, well-being, and social harmony.
- Encourage students to apply and integrate IKS principles with modern knowledge to develop innovative, sustainable, and context-specific solutions.

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

CO	Course Outcome	Knowledge Level (K)#
CO1	Explain the philosophical foundations, evolution, and key principles of Indian Knowledge Systems.	
CO2	Analyze the multidisciplinary contributions of IKS in areas such as mathematics, astronomy, medicine, ecology, governance, and the arts.	
CO3	Analyze the relevance of traditional knowledge to contemporary challenges in sustainability, health, well-being, and social harmony.	
CO4	Compare Indian Knowledge Systems with modern scientific approaches and evaluate their complementarities, differences, and contemporary relevance.	
CO5	Apply IKS principles to develop innovative and sustainable solutions, practices, or projects integrating indigenous knowledge with modern contexts.	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1														
CO2														
CO3														
CO4														
CO5														

(Please fill the above with Levels of Correlation, viz., L, M, H)

UNIT	CONTENTS	Contact Hours
UNIT – 1	Introduction to IKS: Definition and characteristics of Indian Knowledge Systems; Historical evolution from ancient times to the 18th century CE; Impact of colonial education policies and the need for revisiting traditional knowledge; Traditional educational institutions: Gurukuls, Pathshalas, Takshashila, and Nalanda. Local heritage sites and their relevance. Introduction to Ancient Indian Mathematics and Astronomy: Mathematics: Ancient numeral systems and mathematical concepts; Logic: Indian logic systems and epistemology. Overview of Indian astronomy; celestial coordinate systems and calendar systems Astronomical Instruments (Yantras)- Application of Physics and Chemistry	
UNIT – 2	Introduction to Health and Medicine: In-depth study of Ayurveda, its diagnostic methods, and therapeutic practices; Understanding the holistic approach to health encompassing physical, mental, and spiritual well-being; Application of Ayurveda	

	principles in contemporary healthcare systems; Ayurvedic perspectives on health, including dietary regimes, disease management, elements of wellness—and their intersections with botany and medical science	
UNIT – 3	Introduction to Ancient Governance and Public Administration: Analysis of ancient Indian governance models and administrative structures; Insights from texts like Arthashastra on statecraft and economics; Relevance of traditional governance principles in modern administrative practices; Concept of Kingship; duties and responsibilities of a monarch; Three-tier political system: Dharmadanda, Rajdanda, Nya¯yadanda; Law & administration, crime suppression, defence systems, foreign policy; Concept of wealth, ownership, and distribution; Kautilya's "Saptanga" model (seven sources of income); Taxation, savings, and expenditure in ancient Indian economy	
UNIT – 4	Introduction to Ancient Arts and Culture: Study of traditional Indian art forms, including sculpture, dance, and music; Understanding the cultural significance and philosophical underpinnings of these art forms; Promotion of cultural heritage through education and practice; Temple architecture overview; Cave and monolithic architecture; Architectural styles: Chalukya, Pallava, Chola, Hoysala, Mauryan, Vijayanagara; Buddhist and Jain art & architectural heritage	
UNIT – 5	Introduction to Ancient Environmental Sustainability: Traditional knowledge systems related to agriculture, water management, and ecological balance; Application of sustainable practices derived from ancient texts and practices; Integration of IKS in contemporary environmental conservation efforts.	

Learning Resources

Text Books:

1. Introduction to Indian Knowledge System: Concepts and Applications by B. Mahadevan et al.
2. Indian Knowledge System by Kapil Kapoor and Avadhesh Kumar Singh
3. Traditional Knowledge System in India by Amit Jha

Online Courses:

1. Introduction to Indian Knowledge Systems by Bharat Vidya
2. IKS Concepts and Applications in Engineering by CESE, IIT Roorkee
3. IKS – Humanities and Social Sciences by Bhaktivedanta Research Center