B.Tech. – Computer Science and Engineering

Programme 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:** Graduates will have successful careers in computer science and related fields, demonstrating strong technical competence.
- **PEO2:** Graduates will exhibit effective communication, teamwork, and leadership skills in professional settings.
- **PEO3:** Graduates will pursue continuous learning and adapt to emerging technologies and industry trends.
- **PEO4:** Graduates will contribute to society through ethical practice and innovation in computing

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:** The graduate will be able to design, develop and deploy software systems and applications using modern technologies and programming paradigms to solve real-world problems.
- **PSO2:** The graduate will be able to design and implement database systems, data analytics solutions, and AI/ML models to extract insights from large data sets and make informed decisions.
- **PSO3:** The graduate will be able to design and build secure, scalable, and reliable network systems and cyber security solutions to protect critical information and infrastructure.

B.Tech. – I Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	BSC		Physics for Advanced Computing	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		Problem Solving using C	3	0	0	3
5	ESC		Digital Logic Design	2	0	0	2
6	BSC		Physics for Advanced Computing Lab	0	0	3	1.5
7	BSC		Applied Chemistry for Emerging Technologies Lab	0	0	3	1.5
8	ESC		Problem Solving using C Lab	0	0	3	1.5
9	ESC		IT Workshop	0	0	3	1.5
10	MNC		Sports and Yoga	0	0	2	0
		Total		**14**	**0**	**14**	**20**

B.Tech - I Year II Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	HSM		English for Technical Communication	3	0	0	3
2	BSC		Discrete Mathematical Structures	3	0	0	3
3	ESC		Data Structures	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	HSM		English for Technical Communication and MOODLES Lab	0	0	3	1.5
10	ESC		Python Programming Lab	0	0	2	1.0
11	MNC		Environmental Studies	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		Differential Equations and Vector Calculus	3	0	0	3
2	HSM		Engineering Economics and Financial Analysis	3	0	0	3
3	PCC		Object Oriented Programming through JAVA	3	0	0	3
4	PCC		Design and Analysis of Algorithms	3	0	0	3
5	PCC		Principles of Software Engineering	3	0	0	3
6	PCC		Computer Organization and Architecture	3	0	0	3
7	PCC		Object Oriented Programming through JAVA Lab	0	0	3	1.5
8	PCC		Algorithm Analysis 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 Networks	3	0	0	3
3	PCC		Operating Systems	3	0	0	3
4	PCC		Database Management Systems	3	0	0	3
5	PCC		Theory of Computation	3	0	0	3
6	PEC		**Professional Elective -1** 1. Any course from Pool 1 2. Any course from Pool 2 3. Any course from Pool 3 4. Any course from Pool 4 5. Any course from Pool 5 6. Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
7	PCC		Computer Networks and Operating Systems Lab	0	0	3	1.5
8	PCC		Database Management Systems 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 – 08 weeks duration during summer vacation							

B.Tech.- III Year I Semester

S. No.	Course Category	Course Code	Title		T	P	Credits
1	PCC		Data Warehousing and Mining	3	0	0	3
2	PCC		Full Stack Development	3	0	0	3
3	PCC		Compiler Design	3	0	0	3
4	PEC		**Professional Elective-2** 1. Any course from Pool 1 2. Any course from Pool 2 3. Any course from Pool 3 4. Any course from Pool 4 5. Any course from Pool 5 6. Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
5	PEC		**Professional Elective-3** 1 Any course from Pool 1 2 Any course from Pool 2 3 Any course from Pool 3 4 Any course from Pool 4 5 Any course from Pool 5 6 Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
6	PCC		Data Mining Lab	0	0	3	1.5
7	PCC		Full Stack Development Lab	0	0	3	1.5
8	OEC		Open Elective-1	0	0	2	1
9	IPW		Evaluation of industry Internship/ CSP / Mini Project*	0	0	0	2
10	MNC		Indian Knowledge System	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. IT Workshop
2. Any 4 week Skill Oriented/ Fine Arts/ Leadership Courses (online)

B.Tech.- III Year II Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	PCC		Cryptography and Network Security	3	0	0	3
2	PCC		Distributed Systems	3	0	0	3
3	PCC		Machine Learning	3	0	0	3
4	PEC		**Professional Elective-4** 1 Any course from Pool 1 2 Any course from Pool 2 3 Any course from Pool 3 4 Any course from Pool 4 5 Any course from Pool 5 6 Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
5	PEC		**Professional Elective-5** 1 Any course from Pool 1 2 Any course from Pool 2 3 Any course from Pool 3 4 Any course from Pool 4 5 Any course from Pool 5 6 Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
6	OEC		Open Elective-2 / SWAYAM / NPTEL 8 week MOOC	2	0	0	2
7	PCC		Cryptography and Network Security Lab	0	0	3	1.5
8	PCC		Machine Learning Lab	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 - 08 weeks duration during summer vacation							

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

1. Data Structures (L-T-P-C 2-0-0-2)
2. Principles of Data Base Management Systems (L-T-P-C 2-0-0-2)
3. Any 8 week Skill Oriented/ Fine Arts/ Leadership Courses (online)

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

1. Data Structures Lab (L-T-P-C 0-0-2-1)
2. Principles of Data Base Management Systems Lab (L-T-P-C 0-0-2-1)
3. Any 4 week Skill Oriented/ Fine Arts/ Leadership Courses (online)

B.Tech. – IV Year I Semester

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	PCC		Artificial Intelligence	3	0	0	3
2	PCC		Quantum Computing	3	0	0	3
3	PEC		**Professional Elective-6** 1. Any course from Pool 1 2. Any course from Pool 2 3. Any course from Pool 3 4. Any course from Pool 4 5. Any course from Pool 5 6. Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
4	PEC		**Professional Elective-7** 1. Any course from Pool 1 2. Any course from Pool 2 3. Any course from Pool 3 4. Any course from Pool 4 5. Any course from Pool 5 6. Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
5	PEC		**Professional Elective-8** 1. Any course from Pool 1 2. Any course from Pool 2 3. Any course from Pool 3 4. Any course from Pool 4 5. Any course from Pool 5 6. Any 12 week online course SWAYAM/ NPTEL/ Wiser course/ IBM course	3	0	0	3
6	PEC		**Professional Elective-9**	0/1	0	2/0	1
7	OEC		Open Elective-4	2	0	0	2
8	OEC		Open Elective-5	0	0	2	1
9	IPW		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 (L-T-P-C 0-0-2-1)
2. Any 4 week/ 1 credit NPTEL/ SWAYAM/ SWAYAM Plus Course

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

1. Object Oriented Programming Through JAVA (L-T-P-C 2-0-0-2)
2. Any 8 week Skill Oriented/ Fine Arts/ Leadership Courses (online)

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

1. Object Oriented Programming Through JAVA Lab (L-T-P-C 0-0-2-1)
2. Any 4 week Skill Oriented/ Fine Arts/ Leadership Courses (online)

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. Principles of Software Engineering
2. Any 12 week online course NPTEL/ SWAYAM/ SWAYAM Plus/ Wiser course/ IBM course

Courses offered under Open Elective-7 (for other department students):

1. Human Computer Interaction

2 Any 8 week online course NPTEL/ SWAYAM/ SWAYAM Plus/ Wiser course/ IBM course

Professional Electives

Pool 1: AI&ML	Pool 2: Software Engineering
1 Neural Networks and Fuzzy Logic 2 Deep Learning 3 Optimization Techniques in Machine Learning 4 Natural Language Processing 5 Computer Vision 6 Reinforcement Learning 7 Generative AI 8 Building LLMs 9 Responsible and Ethical AI	1. Human Computer Interaction 2. Object Oriented Analysis and Design 3. Agile Methodologies 4. Software Testing Methodologies 5. Software Architecture and Design Patterns 6. Principles of Secure Coding 7. Software Project Management 8. Digital Marketing 9. DevOps
Pool 3: Cyber Security	**Pool 4: Data Science & Analytics**
1 Cyber Security 2 Cyber Laws and Security Policies 3 Web Application Security 4 Ethical hacking 5 Digital Forensics 6 Incident response and Forensics 7 Block Chain Technology 8 Cyber Physical Systems 9 Cloud Security	1. Data Science 2. Data Visualization 3. Information Retrieval Systems 4. Cloud Computing 5. NoSQL Databases 6. Big Data Analytics 7. Business Intelligence 8. Social Media Analytics 9. Stream Data Processing and Analytics
Pool 5: Computer Networks	
1 TCP/IP Protocol Suite 2 Network Programming 3 Mobile Computing 4 Adhoc and Sensor Networks 5 Internet of Things 6 Software Defined Networks 7 Vehicular Adhoc Networks 8 Social Network Analysis 9 Edge and Fog Computing	

Note:

1. While choosing the online courses, the course title/ content should not be replicated with any other course and also student is permitted to complete the course one semester in advance.
2. The Online course platform should have the proctored examination (internal, external assessment and credits).
3. Students may be permitted to choose 16 week/ 4 credit online course in case of non availability of required credit/ non replicated course, but it may be considered as a 3 credit course.

Suggested Courses for **Minor(s) in CSE** – 18 Credits

1. Computer Organization (II-I) (3-0-0-3)
2. Operating Systems (II-II) (3-0-0-3)
3. Object Oriented Analysis and Design (III-I) (3-0-3-4.5)
4. Web Technologies (III-II) (3-0-3-4.5)
5. Computer Networks (IV-I) (3-0-0-3)
6. Any 4/8/12/16 week online course/(s) NPTEL/ SWAYAM/ SWAYAM Plus/Wiser course/IBM course with prior approval BoS.

Suggested Courses for **Minor(s) in Quantum Computing** – 18 Credits

1. Survey of Quantum Technologies and Application (II-I) (3-0-0-3)
2. Foundations of Quantum Technologies (II-II) (3-0-0-3)
3. Introduction to Quantum Computation (III-I) (3-0-0-3)
4. Engineering Foundations of Quantum Technologies (III-II) (3-0-0-3)
5. Quantum Cyber Security (IV-I) (3-0-0-3)
6. Quantum Machine Learning (IV-I) (3-0-0-3)
7. Any 4/8/12/16 week online course/(s) NPTEL/ SWAYAM/ SWAYAM Plus/Wiser course/IBM course with prior approval BoS.

Suggested Courses for **Honor's** – 18 Credits

1. Advanced Operating Systems (III-I) (3-0-0-3)
2. Time Series Analysis and Forecasting (III-I) (3-0-0-3)
3. Recommender Systems (III-II) (3-0-0-3)
4. Augmented Reality and Virtual Reality (III-II) (3-0-0-3)
5. High Performance Computing (IV-I) (3-0-0-3)
6. Nature Inspired Computing (IV-I) (3-0-0-3)
7. Any 4/8/12/16 week online course/(s) NPTEL/ SWAYAM/ SWAYAM Plus/Wiser course/IBM course with prior approval BoS.

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 – R26XXXX	PHYSICS FOR ADVANCED COMPUTING	BSC	L-T-P 3-0-0	CREDITS: 3

Pre-requisite:

Course Objectives: The main objectives of the course are to

1. Equip students the concepts of interference, diffraction and polarization and their applications in optics and communication systems.
2. 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. Explain various electron theories and interpret the effect of temperature on Fermi-Dirac distribution functions and provides basic knowledge of quantum mechanics for quantum computing applications.
4. Familiarize students with semiconductors and magnetic materials and their applications.
5. Impart knowledge on low temperature physics and superconductivity and their application

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

CO		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 such as 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

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 11	PSO 1	PSO 2	PSO 3
CO1	H	H	M	M	M	L	–	–	L	–	M	H	M	–	L
CO2	H	H	M	M	H	M	–	L	M	–	M	H	M	–	H
CO3	H	H	M	M	M	M	–	L	M	–	H	H	H	H	M
CO4	H	H	M	H	M	M	L	L	M	L	M	H	M	M	H
CO5	H	H	H	M	M	H	L	L	M	L	H	H	M	L	M

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

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	10

	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 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 andtime-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	45

Text Books:

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

Reference Books:

1. D. K. Bhattacharya & Poonam Tandon, Engineering Physics, Oxford Press, 2015
2. B. K. Pandey & S. Chaturvedi, Engineering Physics, Cengage Learning, 2021
3. S. O. Pillai, Solid State Physics, New Age International (2019)
4. Nielsen & Chuang, Quantum Computation & Quantum Information, Cambridge (2010)

Web References:

1. NPTEL – nptel.ac.in
2. MIT OpenCourseWare – ocw.mit.edu
3. HyperPhysics – hyperphysics.phy-astr.gsu.edu
4. PhET Simulations – phet.colorado.edu
5. Library of Congress – loc.gov/rr/scitech/selected-internet/physics.html

COURSE CODE – R26XXXX	APPLIED CHEMISTRY FOR EMERGING TECHNOLOGIES	BSC	L-T-P 3-0-0	CREDITS : 3

Pre-requisite: **Chemistry (Higher Secondary level).**

Course Objectives: The main objectives of the course ar eto

- 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 biosensor and utilize in computer applications.
- Explore the chemistry, synthesis methods, and properties of nanomaterials used in nanoelectronics, quantum computing, and sustainable electronics.

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

		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, supercapacitors, 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 nanoelectronics.	K5, K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO 10	PO 11	PSO 1	PSO2	PSO3
CO1	H	M	L	M	L	-	-	-	-	-	M	M	L	-
CO2	H	H	M	M	M	-	-	-	-	-	M	M	M	-
CO3	H	H	M	M	M	M	L	-	-	L	M	L	L	M
CO4	M	H	H	M	M	M	L	-	-	L	M	M	L	H
CO5	H	M	H	H	H	M	L	L	-	L	H	M	M	M

(Please fill the above with Levels of Correlation, viz., L, M, 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;	9

	Fundamentals of quantum mechanics: Schrödinger wave equation, significance of Ψ and Ψ^2; quantum numbers; particle in a one-dimensional box. **Molecular orbital theory:** bonding in homonuclear and heteronuclear diatomic molecules; energy-level diagrams of O_2, N_2 and CO, 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 (SiO_2) 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, Nb_3Sn), 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 for (working, advantages, applications) **Super capacitors:** EDLC, pseudo-capacitors, hybrid super capacitors 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 methods [chemical vapor deposition (CVD), sol-gel process], Applications. **Quantum dots:** synthesis, properties, applications in displays and real time applications. **Metallic nanoparticles:** preparation and applications of gold, silver, and copper nanoparticles. Applications in nanoelectronics, interconnects, sensors (optical and biosensors), memory devices.	9

	Spintronics materials: magnetic nanoparticles, topological insulators	
	Total	**46**

Text Books:

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. Poole, C. P., & Owens, F. J. (2003). Introduction to nanotechnology. John Wiley & Sons.
4. Sharma, B. K., & Gaur, H. (2019). Engineering chemistry. Krishna Prakashan Media.
5. Dara, S. S., & Umare, S. S. (2018). Engineering chemistry (15th ed.). S. Chand Publishing.

Web References:

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

COURSE CODE – R26XXXX	LINEAR ALGEBRA AND CALCULUS	BSC	L-T-P 3-0-0	CREDITS: 3

Pre-requisite:

Course Objectives:

- The main objective of this course is to provide the learner with a 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: *At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Compute rank and inverse of a matrix and hence solves linear system of equations both homogeneous and non-homogeneous.	K2,K3,K5
CO2	Calculate eigenvalues and eigenvectors and hence orthogonalize the given matrix and represent the given quadratic form into canonical form.	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

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO4	PO5	PO 6	PO7	PO8	PO 9	PO 10	PO11	PSO1	PSO 2	PSO3
CO1	**H**	**H**	**M**	**M**	**M**	**–**	**–**	**–**	**L**	**–**	**M**	**H**	**M**	**M**
CO2	**H**	**H**	**M**	**M**	**M**	**–**	**–**	**–**	**L**	**–**	**M**	**H**	**M**	**M**
CO3	**H**	**H**	**M**	**M**	**M**	**M**	**–**	**–**	**M**	**–**	**M**	**M**	**H**	**L**
CO4	**H**	**H**	**M**	**H**	**M**	**M**	**–**	**L**	**M**	**–**	**H**	**H**	**H**	**M**
CO5	**H**	**H**	**M**	**H**	**M**	**M**	**–**	**–**	**M**	**–**	**H**	**M**	**H**	**M**

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

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 – 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).	
	Total	

Text Books:

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

Reference Books:

1. Thomas Calculus, George B. Thomas, Maurice D. Weir and Joel Hass, Pearson Publishers, 2018, 14th Edition.
2. Higher Engineering Mathematics, B.V.Ramana, McGraw Hill Education,2017.
3. Advanced Modern Engineering Mathematics, Glyn James, Pearson publishers, 2018, 5th Edition.
4. Advanced Engineering Mathematics, Micheael Greenberg, Pearson publishers, 9th Edition.
5. Higher Engineering Mathematics, H. K Das, Er. Rajnish Verma, S. Chand Publications, 2014, Third Edition (Reprint 2021).
6. Advanced Engineering Mathematics by H. K Dass, S. ChandPublications,2022, Twenty Two Edition Reprint (2022).
7. Numerical Python: A Practical Techniques Approach for Industry, Robert Johansson, APress, 2015.

Web References:

1. https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors
2. https://en.wikipedia.org/wiki/System_of_linear_equations
3. https://www.math.hmc.edu/calculus/tutorials/eigenstuff/
4. https://en.wikipedia.org/wiki/Quadratic_form
5. https://en.wikipedia.org/wiki/Calculus
6. https://en.wikipedia.org/wiki/Partial_derivative
7. https://www.whitman.edu/mathematics/calculus_online/section14.03.html
8. https://en.wikipedia.org/wiki/Multiple_integral

COURSE CODE – R26XXXX	PROBLEM SOLVING USING C	ESC	L-T-P 3-0-0	CREDITS: 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

	PO 1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	H	H	M	M	M	L	-	-	M	L	M	H	M	L
CO2	H	H	H	M	H	-	L	M	M	-	M	H	M	L
CO3	H	H	H	M	H	M	-	M	M	L	M	H	H	M
CO4	H	H	H	H	H	M	L	M	L	-	H	H	H	M
CO5	H	M	H	M	H	-	M	-	H	M	H	H	M	M

(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:** problem Decomposition, series/pattern Recognition, Abstraction, Algorithm Design; representing solutions as	12

	algorithms, pseudocode, and flowcharts; Program Execution. **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. **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.	9
UNIT – 4	**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. operations on files.	6
	Total	45

Text Books:

1. Let us C, Yashavant Kanetkar, 16th Edition, BPB Publications, 2018, ISBN: 978-93-8728-449-4
2. C Programming, A Problem Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE, 3rd edition, ISBN-13: 978-0-534-49132-1.
3. Programming in C, Reema Theraja, 3rd edition, Oxford University Press, 2023, ISBN-13: 9789354979453

Reference Books:

1.Sumitabha Das, “Computer Fundamentals and Programming”, 18th Edition, McGraw Hill Education, 2018.

2. Ashok N. Kamthane, Amit Ashok Kamthane, "Programming in C", 4th Edition, Pearson Education, 2025. ISBN: 978-9332543633.
3. Pradip Dey, Manas Ghosh, "Computer Fundamentals and Programming in C", 2nd Edition, Oxford University Press, 2013. ISBN: 978-0198068842.
4. Brian W. Kernighan & Dennis M. Ritchie, The C Programming Language, 2nd Edition, Pearson Education / Prentice Hall India, 1988.

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 – R26XXXX	DIGITAL LOGIC DESIGN	ESC	L-T-P 2-0-0	CREDITS: 2

Pre-requisite:

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

	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
CO1	H	M	M	L	M	-	L	L	M	-	M	M	L	M
CO2	H	H	H	M	H	L	-	M	M	L	M	H	L	M
CO3	H	H	H	M	H	M	L	M	M	-	M	H	L	H
CO4	H	H	M	H	H	M	-	M	M	L	H	M	L	H
CO5	H	H	H	H	H	M	-	M	M	M	H	H	L	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 – R26XXXX	PHYSICS FOR ADVANCED COMPUTING LAB	BSC	L-T-P 0-0-3	CREDITS: 1.5

Pre-requisite:

Course Objectives: (3 to 4)

Course Outcomes (4 to 6): *At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Apply principles of interference, diffraction, and polarization to measure physical dimensions and optical properties using specialized instruments.	K3
CO2	Analyze electromagnetic induction, magnetic field distributions, and material hysteresis characteristics through structured experimentation.	K4
CO3	Evaluate semiconductor parameters, energy band gaps, thermal coefficients, and quantum phenomena using electronic setups.	K5
CO4	Examine mechanical oscillations, wave mechanics on stretched strings, and pendulum dynamics to derive fundamental physical constants	K3
CO5	Interpret empirical observations, execute graphical modeling, and perform rigorous error analysis to validate theoretical physics laws.	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO4	PO5	PO 6	PO7	PO8	PO 9	PO 10	PO11	PSO1	PSO 2	PSO3
CO1	H	M	-	H	M	-	-	L	M	-	L	-	-	-
CO2	H	M	-	H	M	-	-	L	M	-	L	-	-	-
CO3	H	M	-	H	M	-	-	L	M	-	L	-	L	-
CO4	H	M	-	H	M	-	-	L	M	-	L	-	-	-
CO5	M	H	-	H	H	-	L	M	H	-	M	-	L	-

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

LIST OF EXPERIMENTS

(Any 10 of the following listed experiments)
(Out of which any **TWO** experiments may be conducted in virtual mode)

1. Determination of thickness of thin object by wedge method.
2. Determination of radius of curvature of a given plano convex lens by forming Newton's rings.
3. Determination of wavelengths of different spectral lines in mercury spectrum using diffraction grating in normal incidence configuration.
4. Determination of resolving power of telescope.
5. Determination of dielectric constant of given capacitor by 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 wavelength of Laser and grating element.

9. Estimation of Planck's constant using photoelectric effect.
10. Determination of the Resistivity of semiconductor by four probe method.
11. Determination of the energy gap of a semiconductor using 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 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. S. Balasubramanian, M.N. Srinivasan "A Text Book of Practical Physics"- S Chand Publishers, 2017.

V-Lab: vlab.co.in

COURSE CODE – R26XXXX	APPLIED CHEMISTRY FOR EMERGING TECHNOLOGIES LAB	BSC	L-T-P 0-0-3	CREDITS: 1.5

Pre-requisite: **Concurrent with Applied Chemistry for Emerging Technologies (Theory**)

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 (4 to 6): *At the end of the course, student will be able to*

		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	Synthesise and examine nanomaterials and evaluate their optical and adsorption properties for engineering applications.	K4
CO5	Conduct experiments safely following laboratory protocols, and record, interpret and report data working individually and in teams.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

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

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

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 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

COURSE CODE – R26XXXX	PROBLEM SOLVING USING C LAB	ESC	L-T-P 0-0-3	CREDITS: 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.
- 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

	PO1	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
CO1	H	H	M	M	M	L	-	L	M	L	M	H	L	L
CO2	H	H	H	M	H	-	L	M	M	-	M	H	L	L
CO3	H	H	H	M	H	M	-	M	M	-	M	H	M	L
CO4	H	H	H	M	H	M	-	L	M	L	H	H	M	L
CO5	H	H	M	H	H	L	M	-	H	M	H	H	M	L

(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 11: 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 12: 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 13: 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.

Exercise 14: Case Studies:

1. A school needs a simple C program to manage student records. The system should allow a user to input details for multiple students (Name, Roll Number, and marks in 3 subjects), calculate their total and average scores, assign a grade, display the report, and save the data to a text file.

(use Variables & Data Types, Operators, Conditional Statements (if-else), Loops (for, while), Arrays, Functions, Structures, and File Handling concepts).

2. Design a command-line ATM interface where a user can check their account balance, deposit money, withdraw money, or exit. The account balance must securely update across different function calls. (use Switch-Case, Infinite/Conditional Loops (while, do-while), Functions, Pointers (Call by Reference), and User-Defined Input Validation concepts)

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 – R26XXXX	IT WORKSHOP	ESC	L-T-P 0-0-3	CREDITS : 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

	PO 1	PO 2	PO3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PSO 1	PSO 2	PS O3
CO1	**H**	**M**	**M**	**L**	**H**	**-**	**M**	**M**	**L**	**L**	**M**	**M**	**L**	**M**
CO2	**M**	**M**	**L**	**M**	**H**	**L**	**M**	**-**	**M**	**L**	**H**	**M**	**L**	**H**
CO3	**M**	**M**	**L**	**-**	**H**	**M**	**H**	**L**	**L**	**-**	**H**	**M**	**L**	**H**
CO4	**M**	**M**	**M**	**L**	**H**	**L**	**-**	**M**	**H**	**M**	**M**	**H**	**H**	**L**
CO5	**M**	**M**	**M**	**L**	**H**	**M**	**L**	**-**	**H**	**L**	**H**	**H**	**H**	**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.

Exercise 13: Case Studies:

1. Design a professional 1-page resume to apply for internships, jobs, or club positions.
2. Create a neat, color-coded grid showing weekly class timings, lab hours, and study slots.
3. Create a 10-slide presentation to explain a mini-project or seminar topic to the class.
4. Create a marks sheet for multiple subjects, automatically calculate the total and average marks, and display a "Pass" or "Fail" status. (Hint: Using basic formulas (SUM, AVERAGE) and a simple logical condition (=IF(Average>=40, "Pass", "Fail"))
5. Design a single-slide promotional flyer or poster for a college tech fest, workshop, or cultural club event.

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 – R26XXXX	SPORTS AND YOGA	MNC	L-T-P 0-0-2	CREDITS: 0

Course Objective(s): The main objectives of the course are to

- Make the students understand the importance of sound health and fitness principles as they relate to better health.
- Expose the students to a variety of physical and yogic activities aimed at stimulating their continued inquiry about Yoga, physical education, health and fitness.
- Create a safe, progressive, methodical and efficient activity- b a se d plan to enhance improvement and minimize risk of injury.
- develop among students an appreciation of physical activity as a lifetime pursuit and a means to better health.

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

CO1 (Physical & Yogic Practice): Practice Hatha Yoga postures and physical activities to enhance overall physical strength, flexibility, balance, coordination, and body relaxation.

CO2 (Mental Health & Focus): Apply breathing exercises and stress-reduction techniques to increase concentration, decrease anxiety, and manage age- or lifestyle-related psychological challenges for improved academic performance.

CO3 (Movement & Combination Forms): Perform diverse yoga movements and physical routines across various combinations, sequences, and forms.

CO4 (Fitness Assessment & Improvement): Assess personal fitness levels and track improvements in health-related fitness components, including cardiorespiratory endurance, flexibility, and body composition.

CO5 (Biomechanics & Training Principles): Apply biomechanical and physiological principles related to exercise, physical training, and the fitness value of different yogic activities.

CO6 (Nutrition & Safety): Demonstrate an understanding of sound nutritional practices and apply injury prevention principles relevant to yoga and physical fitness.

CO7 (Community & Lifestyle Integration): Identify and engage in opportunities for lifelong participation in sports, yoga, and healthy fitness activities.

Mapping of course outcomes with program outcomes

	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PSO 1	PSO 2	PSO 3
CO1	L	-	-	-	-	H	M	M	L	-	H	-	-	-
CO2	L	-	-	-	-	H	M	L	M	-	H	-	-	-
CO3	L	-	-	-	-	H	M	M	L	-	H	-	-	-
CO4	M	L	-	L	-	H	M	L	M	-	H	-	-	-
CO5	M	M	-	M	L	H	M	L	M	-	H	-	-	-
CO6	L	-	-	-	-	H	H	L	L	-	H	-	-	-
CO7	-	-	-	-	-	H	H	M	M	L	H	-	-	-

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

Module I: Introduction to Physical Education

- Meaning & definition of Physical Education
- Aims & Objectives of Physical Education
- Changing trends in Physical Education

Module II: Olympic Movement

- Ancient & Modern Olympics (Summer & Winter)
- Olympic Symbols, Ideals, Objectives & Values
- Awards and Honours in the field of Sports in India (Dronacharya Award, Arjuna Award, Dhayanchand Award, Rajiv Gandhi Khel Ratna Award etc.)

Module III: Physical Fitness, Wellness & Lifestyle

- Meaning & Importance of Physical Fitness & Wellness
- Components of Physical fitness
- Components of Health related fitness
- Components of wellness
- Preventing Health Threats through Lifestyle Change
- Concept of Positive Lifestyle

Module IV: Fundamentals of Anatomy & Physiology in Physical Education, Sports and Yoga

- Define Anatomy, Physiology & Its Importance
- Effect of exercise on the functioning of Various Body Systems. (Circulatory System, Respiratory System, Neuro-Muscular System etc.)

Module V: Kinesiology, Biomechanics & Sports

- Meaning & Importance of Kinesiology & Biomechanics in Physical Edu. & Sports
- Law of Motion & its application in sports.
- Friction and its effects in Sports.

Module VI: Postures

- Meaning and Concept of Postures.
- Causes of Bad Posture.
- Advantages & disadvantages of weight training.
- Concept & advantages of Correct Posture.
- Common Postural Deformities Knock Knee; Flat Foot; Round Shoulders; Lordosis, Kyphosis, Bow Legs and Scoliosis.
- Corrective Measures for Postural Deformities

Module VII: Yoga

- Meaning & Importance of Yoga
- Elements of Yoga
- Introduction - Asanas, Pranayama, Meditation & Yogic Kriyas
- Yoga for concentration & related Asanas (Sukhasana; Tadasana; Padmasana & Shashankasana)
- Relaxation Techniques for improving concentration - Yog-nidra

Module VIII: Yoga & Lifestyle

- Asanas as preventive measures.
- Hypertension: Tadasana, Vajrasana, Pavan Muktasana, Ardha Chakrasana, Bhujangasana, Sharasana.

- Obesity: Procedure, Benefits & contraindications for Vajrasana, Hastasana, Trikonasana, Ardh Matsyendrasana.
- Back Pain: Tadasana, Ardh Matsyendrasana, Vakrasana, Shalabhasana, Bhujangasana.
- Diabetes: Procedure, Benefits & contraindications for Bhujangasana, Paschimottasana, Pavan Muktasana, Ardh Matsyendrasana.
- Asthema: Procedure, Benefits & contraindications for Sukhasana, Chakrasana, omukhasana, Parvatasana, Bhujangasana, Paschimottasana, Matsyasana.

Module IX: Training and Planning in Sports

- Meaning of Training
- Warming up and limbering down
- Skill, Technique & Style
- Meaning and Objectives of Planning.
- Tournament Knock-Out, League/Round Robin & Combination.

Module X: Psychology & Sports

- Definition & Importance of Psychology in Physical Edu. & Sports
- Define & Differentiate Between Growth & Development
- Adolescent Problems & Their Management
- Emotion: Concept, Type & Controlling of emotions
- Meaning, Concept & Types of Aggressions in Sports.
- Psychological benefits of exercise.
- Anxiety & Fear and its effects on Sports Performance.
- Motivation, its type & techniques.
- Understanding Stress & Coping Strategies.

Module XI: Doping

- Meaning and Concept of Doping
- Prohibited Substances & Methods
- Side Effects of Prohibited Substances

Module XII: Sports Medicine

- First Aid Definition, Aims & Objectives.
- Sports injuries: Classification, Causes & Prevention.
- Management of Injuries: Soft Tissue Injuries and Bone & Joint Injuries

Module XIII: Sports / Games

Following subtopics related to any one Game/Sport of choice of student out of: Athletics, Badminton, Basketball, Chess, Cricket, Kabaddi, Lawn Tennis, Swimming, Table Tennis, Volleyball, Yoga etc.

- History of the Game/Sport.
- Latest General Rules of the Game/Sport.
- Specifications of Play Fields and Related Sports Equipment.
- Important Tournaments and Venues.
- Sports Personalities.
- Proper Sports Gear and its Importance.

Text Books/References:

1. Modern Trends and Physical Education by Prof. Ajmer Singh.
2. Light On Yoga By B.K.S. Iyengar.
3. Health and Physical Education NCERT (11th and 12th Classes)

COURSE CODE – R26XXXX	ENGLISH FOR TECHNICAL COMMUNICATION	HSM	L-T-P 3-0-0	CREDITS: 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, student will be able to*

		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

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

CO\PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	L	M	–	–	L	M	–	–	L	H	L	L	M	L
CO2	L	L	L	–	L	H	M	M	H	H	M	H	M	M
CO3	L	M	L	L	L	M	L	L	H	H	M	H	M	M
CO4	M	H	M	M	H	L	L	H	M	H	H	M	H	L
CO5	L	M	L	L	L	M	L	L	H	H	H	H	M	M

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, 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 Wrtiting
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 – R26XXXX	DISCRETE MATHEMATICAL STRUCTUES	HSM	L-T-P 3-0-0	CREDITS: 3

Pre-requisite:

Course Objectives: The main objectives of the course are to

- Understand the foundations of mathematical logic and proof techniques.
- Apply techniques and tools for sets, relations and functions.
- Analyze methods and engineering practices of combinatorics and recurrence relations.
- Evaluate alternatives and trade-offs in graph theory and trees.
- Design and document an integrated solution involving algebraic structures and boolean algebra.

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

		Knowledge Level (K)#
CO1	Explain the principles and engineering practices of mathematical logic and proof techniques using appropriate evidence and tools.	K2
CO2	Apply the principles and engineering practices of sets, relations and functions using appropriate evidence and tools.	K3
CO3	Analyze the principles and engineering practices of combinatorics and recurrence relations using appropriate evidence and tools.	K4
CO4	Evaluate the principles and engineering practices of graph theory and trees using appropriate evidence and tools.	K5
CO5	Design and demonstrate the principles and engineering practices of algebraic structures and boolean algebra using appropriate evidence and tools.	K6

#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	M	L	-	L	M	L	H	M	M	L
CO2	H	H	M	M	M	-	L	L	M	-	M	M	M	L
CO3	H	H	M	H	L	L	-	M	M	L	H	M	H	L
CO4	H	H	H	H	M	-	L	M	M	L	H	H	H	H
CO5	H	H	H	M	H	M	-	M	L	M	H	H	H	M

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

UNIT	CONTENTS	Contact Hours
UNIT – 1	MATHEMATICAL LOGIC AND PROOF TECHNIQUES Propositions and logical connectives; truth tables; logical equivalences; normal forms; predicates and quantifiers; inference rules; validity; direct and indirect proofs; contradiction and contrapositive; mathematical induction; strong induction; recursive definitions; proof by cases; applications to program correctness and specification.	
UNIT – 2	SETS, RELATIONS AND FUNCTIONS Set operations and identities; Cartesian products; power sets; relations and their	

	properties; equivalence relations; partial orders; Hasse diagrams; functions; injective, surjective and bijective functions; composition and inverses; cardinality; countable sets; pigeonhole principle; applications to databases, type systems and computing.	
UNIT – 3	COMBINATORICS AND RECURRENCE RELATIONS Counting principles; permutations and combinations; binomial theorem; inclusion-exclusion principle; pigeonhole principle; generating functions concepts; recurrence relations; linear homogeneous recurrences; characteristic roots; nonhomogeneous recurrences; recurrence solving; combinatorial identities; probability foundations; applications to algorithm analysis	
UNIT – 4	GRAPH THEORY Graph terminology; graph representations; degree and connectivity; paths and cycles; Euler and Hamiltonian graphs; bipartite graphs; graph isomorphism concepts; planar graphs; graph coloring; trees and spanning trees; rooted trees; binary trees; traversal; graph algorithms; applications to networks and scheduling.	
UNIT – 5	ALGEBRAIC STRUCTURES AND NUMBER THEORY Binary operations; semigroups, monoids and groups; subgroups; cyclic groups; homomorphisms concepts; rings and fields overview; modular arithmetic; Number Theory: congruences: linear and polynomial congruences; prime numbers: counting primes, pseudo-primes and primality testing; factorization: factorization algorithms; arithmetic functions: multiplicative and additive functions, Euler's phi function, sum and number of divisors functions,; primitive roots and quadratic residues: primitive roots, index arithmetic, quadratic residues, modular square roots; Fermat's last theorem, quadratic fields: primes and unique factorization.	
	Total	

Text Books:

1. Kenneth H. Rosen, Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, McGraw Hill Education (India) Private Limited, 2012.
2. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill, 35th reprint, 2008
3. Graph Theory with Applications to Engineering and Computer Science, Narsingh Deo, PHI, 2010.
4. K. Chandrasekaran, An Introduction to Analytic Number Theory, Springer, 1968

Reference Books:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms (4th ed.). MIT Press.
2. Gustedt, J. (2025). Modern C (3rd ed.). Manning Publications.
3. Strang, G. (2023). Introduction to linear algebra (6th ed.). Wellesley-Cambridge Press.
4. Levin, O. (2023). Discrete mathematics: An open introduction (4th ed.). [Open textbook].

Web References: (Pls include hyperlinks related to NPTEL/Vlabs etc.,)

- https://discrete.openmathbooks.org/
- https://math.mit.edu/~gs/linearalgebra/
- https://www.nptel.ac.in/

COURSE CODE – R26XXXX	DATA STRUCTURES	ESC	L-T-P 3-0-0	CREDITS: 3

Pre-requisite: Computational Thinking and Problem Solving using C

Course Objectives: At the end of the course, the teacher 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

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

(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; Radix sort, Bucket 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; AVL Trees, Red-Black Trees, 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, Kruskall's algoithm); 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:

1. Data Structures and Algorithm Analysis, 4th Edition, Mark Allen Weiss,Pearson.
2. Data Structures: A PseudoCode Approach with C, 2nd Edition, Richard F.Gilberg, &Behrouz A. Forouzon, Cengage.
3. Fundamentals of Data Structures, 2/e, Sartaj Sahni,University Press.
4. Data Structures using C, 2/e, Reema Thareja

Reference Books:

1. Balagurusamy, E. (2025). Data Structures Using C (2nd ed.). McGraw Hill Education India.
2. Thareja, R. (2023). Data Structures Using C (3rd ed.). Oxford University Press India.
3. Deshpande, P. S., & Kakde, O. G. C and data structures. Dreamtech Press.
4. Shukla, R. K. 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 – R26XXXX	PYTHON PROGRAMMING	ESC	L-T-P 2-0-0	CREDITS: 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)
- 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, 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 Python programs to model real-world problems.	K4
CO5	Develop Python applications that perform file processing 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

	PO 1	PO2	PO3	PO4	PO 5	PO 6	PO 7	P O8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	H	H	M	M	H	L	-	M	M	L	H	H	M	L
CO2	H	H	H	M	H	M	L	M	M	-	H	H	H	L
CO3	H	H	H	H	H	L	-	M	M	L	H	H	H	L
CO4	H	H	H	H	H	-	L	L	M	L	H	H	H	M
CO5	H	H	H	H	H	L	-	M	H	M	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.	7
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.	7
UNIT – 3	**FUNCTIONS AND RECURSION** **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.	7
UNIT – 4	**MODULES, LIBRARIES AND FILE HANDLING:** Importing and creating modules, packages and built-in libraries; responsible AI-assisted coding practices. **File Handling:** Reading and writing files; decomposition of a data workflow into read/process/write stages.	7
UNIT – 5	**DATA ANALYSIS** **Data Analysis and Visualization**: Introduction to NumPy, pandas and Matplotlib basics; pattern recognition in real datasets.	7
	Total	**35**

Text Books:

1. Downey, A. B. (2015). Think Python: How to think like a computer scientist (2nd ed.). O'Reilly Media.
2. Core Python Programming, Dr R Nageswara Rao, 3rd ed, Dreamtech.
3. McKinney, W. (2022). Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter (3rd ed.). O'Reilly Media.

Reference Books:

1. J. V. Guttag, Introduction to Computation and Programming Using Python: With Application to Computational Modeling and Understanding Data, 3rd Edition, MIT Press, 2021.
2. Sweigart, A. (2025). Automate the boring stuff with Python: Practical programming for total beginners (3rd ed.). No Starch Press.
3. J. M. Wing, Computational Thinking, Communications of the ACM, 49(3), 33–35, 2006. https://doi.org/10.1145/1118178.1118215
4. Jeeva Jose, "Taming Python by Programming", 2nd Edition, Khanna Book Publishing Co., Delhi, 2019.

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 – R26XXXX	SUSTAINABILITY AND GREEN ENGINEERING	ESC	L-T-P 2-0-0	CREDITS: 2

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

Course Objectives:

- 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: *At the end of the course, student 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 and 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.	K4
CO5	Evaluate alternative engineering solutions and recommend sustainable and green-engineering interventions considering technical, environmental, economic and social criteria.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	**PO2**	**PO 3**	**PO4**	**PO 5**	**PO 6**	**PO7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PSO 1**	**PSO2**	**PSO 3**
CO1	L	L	L	–	–	L	H	L	L	L	–	-	-	-
CO2	H	L	L	L	L	L	H	L	L	L	L	L	-	-
CO3	L	H	L	H	H	L	H	L	L	L	L	L	M	-
CO4	L	H	H	L	H	L	H	L	L	L	L	L	L	-
CO5	L	H	H	L	H	H	H	H	L	L	H	L-	L	-

(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**: 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. **Circular Economy:** Linear versus circular economy; 3R/5R principles; reuse, repair, refurbishment and recycling; industrial symbiosis; EPR; waste-to-resource approaches; circular business models.	6
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, Shantanu Kumar Sahu; Green Technology: An Innovative Solution for Environmental Sustainability, Newredmars Education, 2025

Reference Books:

1. David T. Allen, David R. Shonnard, *Sustainable Engineering: Concepts, Design and Case Studies*, Pearson.
2. David T. Allen, 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 – R26XXXX	DATA STRUCTURES USING C LAB	ESC	L-T-P 0-0-3	CREDITS: 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

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

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

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 11: 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 12: AVL Trees & Hashing

1. Implement AVL Tree with creation, insertion, search and deletion operations along with rotations.
2. Implement a hash table using chaining for collision resolution, with insertion, search and deletion.
3. 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 – R26XXXX	DESIGN THINKING AND INNOVATION	ESC	L-T-P 0-1-3	CREDITS: 2.5

Pre-requisite:

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

		Knowledge Level (K)#
CO1	Explain the principles, processes and mindset of Design Thinking for engineering problem-solving and innovation	K2
CO2	Apply empathy, observation and problem-framing techniques to identify user needs and define engineering problems	K3
CO3	Apply creative and systematic ideation techniques to generate and select innovative solution concepts	K3
CO4	Analyze solution concepts and prototypes using user feedback and appropriate design criteria for iterative improvement	K4
CO5	Analyze the innovation potential of solutions considering value proposition, feasibility, sustainability, societal impact and basic IPR aspects	K4

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO3**	**PO 4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO 9**	**PO 10**	**PO 11**	**PSO 1**	**PSO2**	**PSO3**
CO1	M	M	L	–	–	L	–	–	L	L	M	L	L	L
CO2	M	H	H	L	L	H	M	L	M	M	M	M	M	M
CO3	L	M	H	L	M	L	L	–	H	M	M	H	H	M
CO4	H	H	H	M	H	M	M	L	H	M	M	H	M	L
CO5	L	M	H	L	M	H	H	M	M	H	H	M	M	L

(Please fill the above with Levels of Correlation, viz., L, M, 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 an empathy map • Developing a persona • Preparing a user journey map	
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; 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 – 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 – 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 protection; Responsible innovation; Inclusive and sustainable innovation; Innovation ecosystem and entrepreneurship; Communicating innovative ideas; Innovation pitch **Practical Activities** 1. Classify existing products/services as incremental, disruptive or radical innovations. 2. Identify three innovation opportunities from a campus/community problem. 3. Prepare a **Value Proposition Canvas**. 4. Validate the proposed innovation with potential users. 5. Prepare an introductory **Business Model Canvas**. 6. Identify possible IPR protection. 7. Prepare and deliver a **3-minute Innovation Pitch**.	
	Total	

Text Books:

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 References:

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

COURSE CODE – R26XXXX	AI TOOLS AND APPLICATIONS	ESC	L-T-P 0-1-3	CREDITS: 2.5

Pre-requisite: Basic computer literacy and internet use; IT Workshop

Course Objectives: Basic computer literacy and internet use to

- Familiarize students with the landscape of AI tools and their underlying working principles.
- Develop competence in prompt engineering for effective interaction with generative AI systems.
- Enable students to apply AI tools for data analysis, visualization, and interpretation.
- Train students in using AI-assisted tools for coding, writing, and productivity tasks.
- Install awareness of ethical, legal, and societal considerations in AI tool usage.

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

		Knowledge Level (K)#
CO1	Recall the categories, capabilities, and limitations of major AI tools (text, image, code, data).	K2
CO2	Explain the principles of prompt engineering and how generative AI models process instructions.	K3
CO3	Apply AI tools to solve first-year engineering problems in analysis, writing, and coding tasks.	K3
CO4	Analyze AI-generated outputs for accuracy, bias, and reliability before use.	K4
CO5	Evaluate the ethical, academic-integrity, and societal implications of AI tool use, justifying appropriate usage choices.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO 1	**PO 2**	**PO3**	**PO 4**	**PO 5**	**PO 6**	**PO7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	M	-	-	-	H	L	-	-	-	-	L	-	M	-
CO2	M	M	-	-	H	L	-	L	-	M	L	M	M	-
CO3	H	H	M	M	H	L	L	-	L	M	M	H	H	-
CO4	M	H	M	H	H	M	M	H	L	M	M	-	H	-
CO5	L	M	M	M	M	L	H	H	M	H	M	L	L	L

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

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

- **Computer System with Internet Connectivity** — desktop/laptop (one per student or pair) with a modern web browser (Chrome/Firefox/Edge).
- **Conversational AI Assistants** — e.g., Claude, ChatGPT, or Gemini (free tier), used across most exercises for prompting, writing, and Q&A tasks.
- **AI-Assisted Productivity & Content Tools** — an AI image-generation tool, a writing/productivity assistant (e.g., Grammarly, Microsoft Copilot, Notion AI), and a word processor (MS Word/Google Docs) for notes, reports, and citation work.

- **AI-Assisted Spreadsheet & Data-Analysis Tools** — spreadsheet software with AI features (e.g., MS Excel with Copilot, Google Sheets) and an AI data-analysis tool (e.g., a chatbot with file-upload/code capability, or Kaggle/Google Colab notebooks).
- **AI Coding Assistant with a C/Python Environment** — an AI coding assistant (e.g., GitHub Copilot or a conversational assistant) paired with a compiler/IDE (GCC/Code::Blocks for C; Python 3.x with IDLE/Jupyter/VS Code) to run and verify generated code.
- **Sample Datasets and Email Access** — small instructor-provided CSV/XLSX datasets (e.g., student marks) for the data exercises, and a web-based email client for the productivity/email-drafting exercise.

Unit 1: Introduction to AI and the AI Tools Landscape

1. What is AI, Machine Learning, and Generative AI- basic definitions and how they relate to one another.
2. Core categories of AI tools: conversational/text, code assistants, data-analysis tools, image-generation tools, and writing/productivity tools.
3. How large language models process and generate responses (conceptual, non-mathematical overview).
4. Criteria for selecting the right AI tool for a given task — accuracy needs, cost, data sensitivity, and output format.
5. Setting up accounts and workspaces; understanding free-tier versus paid-tier capabilities and limitations.

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

1. Anatomy of a good prompt — context, clear instruction, desired format, and constraints.
2. Zero-shot, few-shot, and chain-of-thought prompting techniques and when to use each.
3. Role-based and context-setting prompts (e.g., asking the AI to respond as a reviewer, tutor, or domain expert).
4. Iterative prompt refinement - diagnosing a poor AI response and rewriting the prompt to fix it.
5. Using generative AI for brainstorming, summarizing, and drafting technical content.

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

1. AI-assisted spreadsheet and dataset analysis using natural-language commands.
2. Natural-language querying of datasets to extract summary statistics and insights.
3. AI tools for generating charts, graphs, and visual summaries from tabular data.

4. Interpreting AI-generated statistical summaries correctly and recognizing their limitations.
5. Identifying and correcting errors or misinterpretations in AI-generated data analysis.

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

1. AI coding assistants — code generation, explanation, and debugging support for simple programs.
2. AI writing assistants for reports, emails, and technical documentation.
3. AI tools for scheduling, note-taking, and meeting/lecture summarization.
4. AI-assisted presentation and slide-outline generation from a topic prompt.
5. Integrating AI tools into an efficient day-to-day academic and engineering workflow.

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

1. Bias and hallucination in AI outputs, and techniques for detecting and fact-checking them.
2. Academic integrity and appropriate citation of AI assistance in coursework and submissions.
3. Data privacy and security considerations, including obligations under India's Digital Personal Data Protection Act, 2023 (DPDP Act), when uploading personal or institutional data.
4. Copyright and intellectual property considerations in AI-generated text and image content.
5. Emerging trends — multimodal AI and AI agents — and their implications for future engineering practice.

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.

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 – R26XXXX	ENGLISH FOR TECHNICAL COMMUNICATION AND MOODLES LAB	HSM	L-T-P 0-0-3	CREDITS: 1.5

Part – A

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

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
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 – R26XXXX	PYTHON PROGRAMMING LAB	ESC	L-T-P 0-0-2	CREDITS: 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)
- 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, 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 Python programs to model real-world problems.	K4
CO5	Develop Python applications that perform file processing 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

	PO 1	PO2	PO3	PO4	PO 5	PO 6	PO 7	P O8	PO 9	PO 10	PO 11	PS O1	PS O2	PS O3
CO1	H	M	M	-	H	-	-	L	L	-	M	H	M	L
CO2	H	H	H	M	H	-	L	L	L	-	M	H	H	L
CO3	H	H	H	M	H	-	-	M	L	-	H	H	H	M
CO4	H	H	H	M	H	-	L	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 11: 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 12: 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 13: Python Libraries – Numpy, Pandas , Matplotlib

1. Write a NumPy program to generate a matrix product of two arrays.

2. Write a NumPy program to create a random array with 1000 elements and compute the average, variance, standard deviation of the array elements.

3. Demonstrate how to download dataset and how to create DataFrame
   i. Write a Pandas program to get the first 3 rows of a DataFrame
   ii. Write a Pandas program to select the specified columns and rows from a given data frame
   iii. Write a Pandas program to select the rows where the score is missing, i.e. is NaN.
   iv. Write a Pandas program to insert a new column in existing DataFrame.
4. Write a Python programming to display a bar chart using different color for each bar
5. Write a Python programming to create a pie chart with a title.

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 – R26XXXX	ENVIRONMENTAL STUDIES	MNC	L-T-P 2-0-0	CREDITS: 0

Pre-requisite:

Course Objectives:

1. To provide core foundational knowledge on natural resources, ecosystems, and their functional balances.
2. To understand the causes, effects, and mitigation strategies of various environmental pollutions.
3. To evaluate the socio-economic, ethical, and legal frameworks governing environmental protection.
4. To mainstream sustainable development, green engineering practices, and climate action into technical designs.

Course Outcomes (4 to 6): *At the end of the course, student will be able to*

		Knowledge Level (K)#
CO1	Explain the structure, function, and value of ecosystems and natural resources, and assess threats to global biodiversity.	
CO2	Analyze environmental pollution systems and apply scientific principles to design basic control and waste management methods.	
CO3	Evaluate socio-environmental issues, climate change mechanisms, and sustainable urbanization models.	
CO4	Interpret national/international environmental legislations, green chemistry principles, and professional ethics	
CO5	Integrate environmental sustainability aspects into engineering problem-solving and project designs	

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO 2	PO 3	PO4	PO 5	PO6	PO 7	PO 8	PO 9	PO10	PO1 1	PSO 1	PSO 2	PS O3
CO1	L	M	-	L	-	H	L	-	-	-	L	L	M	-
CO2	M	M	M	M	L	H	L	-	-	L	L	L	M	L
CO3	L	M	L	L	-	H	M	L	L	L	M	L	H	L
CO4	L	L	-	-	-	H	H	-	L	L	L	-	-	-
CO5	M	M	H	M	M	H	M	L	L	M	M	L	M	L

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

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Ecosystems, Biodiversity, and Natural Resources (9 Lectures)** **Multidisciplinary nature of environmental studies:** Definition, scope, and importance; Need for public awareness. **Ecosystems:** Concept, structure, and function (producers, consumers,	

	decomposers); Energy flow, food chains, food webs, and ecological pyramids; Case studies of forest, grassland, desert, and aquatic ecosystems. **Biodiversity:** Levels (genetic, species, ecosystem); Hot-spots of biodiversity; India as a mega-diversity nation; Threats to biodiversity (habitat loss, poaching, man-wildlife conflicts) and conservation strategies. **Natural Resources:** Renewable and non-renewable resources; Forest, water, mineral, food, land, and energy resources; Over-exploitation and associated case studies.	
UNIT – 2	**Environmental Pollution and Control Technologies (8 Lectures)** **Environmental Pollution:** Definition, causes, effects, and control measures of Air, Water, Soil, Marine, Thermal, and Noise pollution. **Solid & Hazardous Waste Management:** Municipal solid waste, e-waste, biomedical waste management; E-waste recycling and circular economy models. **Pollution Control Engineering:** Wastewater treatment plants (Primary, secondary, and tertiary treatments); Air pollution control devices (Electrostatic precipitators, scrubbers). **Case Studies:** Industrial accidents, Bhopal gas leak, major oil spills.	
UNIT – 3	**Social Issues, Climate Change, and Sustainable Development (8 Lectures)** **Sustainable Development:** Goals (SDGs), demographic transitions, and urban problems related to energy. **Water Management:** Rainwater harvesting, watershed management, water conservation paradigms. **Global Environmental Issues:** Climate change, global warming, acid rain, ozone layer depletion, and net-zero carbon concepts. **Resettlement and Rehabilitation:** Issues, concerns, and associated case studies (e.g., major dam projects).	
UNIT – 4	**Environmental Legislation, Policies, and Green Technology (7 Lectures)** **Environmental Protection Acts:** Air (Prevention and Control of Pollution) Act, Water Act, Wildlife Protection Act, Forest Conservation Act, and National Green Tribunal Act. **Green Technology & Chemistry:** 12 Principles of Green Chemistry, clean energy sources, green building concepts, and Ecomark labeling. **Environmental Impact Assessment (EIA):** Objectives, process, and role of Information Technology and GIS in environmental management	
	Total	

Text Books:

1. Bharucha, E. (2021). *Textbook of environmental studies for undergraduate courses* (3rd ed.). Orient BlackSwan.
2. Kaushik, A., Kaushik, C. P. (2020). *Perspectives in environmental studies* (7th ed.). New Age International Publishers.
3. Rajagopalan, R. (2016). *Environmental studies: From crisis to cure* (3rd ed.). Oxford University Press.

Reference Books:

1. Cunningham, W. P., & Cunningham, M. A. (2017). *Principles of environmental science: Inquiry and applications* (9th ed.). McGraw-Hill Education.
2. Kiely, G. (2007). *Environmental engineering*. McGraw-Hill Education.
3. Miller, G. T., Spoolman, S. E. (2020). *Living in the environment* (20th ed.) Cengage Learning.

Web References:

1. AICTE. (n.d.). *Model curriculum for environmental studies*. All India Council for Technical Education. https://www.aicte-india.org/