Department of Computer Science Engineering

Vision of the Department:

The Computer Science & Engineering aims at providing continuously stimulating educational environment to its students for attaining their professional goals and meet the global challenges.

Mission of the Department:

DM1: To develop a strong theoretical and practical background across the computer science discipline with an emphasis on problem solving.

DM2: To inculcate professional behaviour with strong ethical values, leadership qualities, innovative thinking and analytical abilities into the student.

DM3: Expose the students to cutting edge technologies which enhance their employability and knowledge.

DM4: Facilitate the faculty to keep track of latest developments in their research areas and encourage the faculty to foster the healthy interaction with industry.

Program Educational Objectives (PEOs):

PEO1: Pursue higher education, entrepreneurship and research to compete at global level.

PEO2: Design and develop products innovatively in computer science and engineering and in other allied fields.

PEO3: Function effectively as individuals and as members of a team in the conduct of interdisciplinary projects; and even at all the levels with ethics and necessary attitude.

PEO4: Serve ever-changing needs of society with a pragmatic perception.

PROGRAMME OUTCOMES (POs):

PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO 6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAMME SPECIFIC OUTCOMES (PSOs):

PSO 1	The ability to apply Software Engineering practices and strategies in software project development using open-source programming environment for the success of organization.
PSO 2	The ability to design and develop computer programs in networking, web applications and IoT as per the society needs.
PSO 3	To inculcate an ability to analyze, design and implement database applications.

R26-COURSE STRUCTURE
(CSE)

I SEMESTER

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	BSC	A2610001	Applied Chemistry in Emerging Technologies	3	0	0	3.0
2	BSC	A2610002	Linear Algebra & Calculus	3	0	0	3.0
3	ESC	A2610003	Computational Thinking & Problem-Solving using C	3	0	0	3.0
4	ESC	A2610004	Digital Logic Design	2	0	0	2.0
5	ESC	A2610005	Design Thinking & Innovation	0	1	3	2.5
6	ESC	A2610006	Basic Electrical and Electronics Engineering	2	0	0	2.0
7	BSC	A2610007	Applied Chemistry in Emerging Technologies Lab	0	0	3	1.5
8	ESC	A2610008	Computational Thinking & Problem-Solving using C Lab	0	0	3	1.5
9	ESC	A2610009	IT and Electronics Workshop	0	0	3	1.5
10	MNC	A2610010	Sports and Yoga or NSS/NCC	0	0	2	0.0
		Total		**13**	**1**	**14**	**20.0**

II -SEMESTER

S.No.	Course Category	Course Code	Title	L	T	P	Credits
1	HSM	A2620001	English for Technical Communication	3	0	0	3.0
2	BSC	A2620002	Physics for Advanced Computing	3	0	0	3.0
3	BSC	A2620003	Differential Equations & Vector Calculus	3	0	0	3.0
4	ESC	A2620004	Data Structures using C	3	0	0	3.0
5	ESC	A2620005	Python Programming	2	0	0	2.0
6	HSM	A2620006	English for Technical Communication and MOODLES Lab	0	0	3	1.5
7	ESC	A2620007	Data Structures using C Lab	0	0	3	1.5
8	ESC	A2620008	AI Tools & Applications	0	1	3	2.5
9	ESC	A2620009	Python Programming Lab	0	0	2	1.0
10	BSC	A2620010	Physics for Advanced Computing Lab	0	0	3	1.5
11	MNC	A2620011	Environmental Science	2	0	0	0.0
		Total		**14**	**1**	**16**	**22**

L	T	P	Cr.
3	0	0	3

B.Tech. (I Sem.) **A2610001-APPLIED CHEMISTRY IN EMERGING TECHNOLOGIES**

Pre-requisite: Basic Knowledge of Chemistry (Higher Secondary level).

Course Objectives:

- Understand the fundamentals of quantum chemistry, including wave-particle duality, and molecular orbital theory relevant to molecular systems.
- 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, fuel cells, and supercapacitors, as energy-storage devices for advanced technologies.
- Design sensors, biosensors and utilize them in modern applications.
- Explore the chemistry, synthesis methods, properties of nanomaterials used in nanoelectronics, quantum computing, and sustainable electronics.

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

CO1: Interpret the principles of quantum chemistry, apply molecular orbital theory to explain the electronic structure and bonding in molecular systems. (**K2** (Understand) & **K3** (Apply))

CO2: Apply the concept of doping to prepare semiconductors, understand properties, applications of superconductors and ion-trap materials in modern electronic devices. (**K3** (Apply) & **K2** (Understand))

CO3: Apply electrochemical principles to assess the performance of batteries, fuel cells and explain the significance of supercapacitors in energy-harvesting applications. (**K3** (Apply) & **K2** (Understand))

CO4: Relate the working principles of sensors, biosensors, smart materials to their applications in modern devices and understand sustainable e-waste management practices. (**K3** (Apply) & **K2** (Understand))

CO5: Summarize nanomaterial synthesis methods, analyse their properties and applications, including spintronic materials. (**K2** (Understand) & **K4** (Analayze))

CO-PO Articulation Matrix:

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

Scale: H - Strong | M - Moderate | L - Slight | - No Correlation.

UNIT I: Quantum Chemistry **[10 hrs]**

Fundamentals of quantum chemistry: Black-body radiation, photoelectric effect, wave-particle duality, de Broglie wave equation, Heisenberg's uncertainty principle.

Fundamentals of quantum mechanics: Schrödinger wave equation (time-independent), significance of Ψ and Ψ^2, particle in a one-dimensional box.

Molecular orbital theory: Bonding in homonuclear and heteronuclear diatomic molecules; energy-level diagrams of O_2, N_2, CO, and HF; π-molecular orbitals of butadiene and benzene; calculation of bond order.

UNIT II: Chemistry of Semiconductors and Superconductors **[10 Hrs]**

Band theory of semiconductors, Intrinsic and extrinsic semiconductors, Preparation of pure and doped semiconductor materials, silicon dioxide (SiO_2) formation and properties, Chemistry of compound semiconductors (GaAs, InP, SiC), Advanced semiconductor materials (Organic semiconductors, Perovskite semiconductors).

Photolithography: Basic principle, and 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, and applications.

UNIT III: Triad of Electrochemical Power [9 Hrs]
Galvanic and electrolytic cells, redox reactions, electrode potential, Nernst equation and problems.
Primary batteries: Sodium-air batteries (construction, working and applications), zinc-air battery.
Secondary batteries: Lithium-ion, sodium-ion batteries (construction, charging-discharging reactions and applications). Battery management systems (main functions, basic working, Battery protection).
Fuel cells: PEMFC, DMFC (working, advantages, applications).
Supercapacitors: EDLC, pseudo-capacitors, hybrid supercapacitors for energy harvesting, applications.

UNIT IV: E-Waste Management, Sensors, Biosensors and Smart Materials [9 Hrs]
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).

UNIT V: Chemistry of nanomaterials [9 Hrs]
Introduction to Nano chemistry.
Carbon nanomaterials: Fullerenes, carbon nanotubes (CNTs), graphene; synthesis methods [chemical vapor deposition (CVD)], Applications.
Quantum dots: Synthesis, properties, applications in displays and real time applications.
Metallic nanoparticles: Preparation (sol-gel process) and applications of gold, silver, and copper nanoparticles. Applications in nanoelectronics, interconnects, sensors (optical and biosensors), memory devices.
Spintronics materials: Magnetic nanoparticles, topological insulators.

Textbooks:

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

Reference Books:

1. Linden, D., & Reddy, T. B. (Eds.). (2001). Handbook of batteries (3rd ed.). McGraw-Hill.
2. Rao, C. N. R., Müller, A., & Cheetham, A. K. (Eds.). (2007). The chemistry of nanomaterials: Synthesis, properties and applications (Vols. 1-2). Wiley-VCH.
3. 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 Resources (APA Format)

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

B.Tech. (I Sem.) A2610002- LINEAR ALGEBRA & CALCULUS

L	T	P	Cr.
3	0	0	3

Course Objectives:

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

Course Outcomes: Upon completion of the course, the student will be able to:

CO1 Analyse the solutions of homogeneous and non-homogeneous systems of linear equations. (K4)

CO2 Evaluate Eigen values and Eigen vectors to reduce quadratic form to canonical form. (K5)

CO3 Estimate the bounds of a given function by mean vale theorems. (K3)

CO4 Apply the knowledge of derivatives in optimizing functions of several variables. (K3)

CO5 Evaluate double and triple integrals in Cartesian and polar coordinates. (K5)

K1- Remembering, K2- Understanding, K3-Applying, K4- Analyzing, K5- Evaluating, K6- Creating

Course Articulation Matrix:

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

H – High, M - Medium, L - Low

UNIT I: Matrices, Eigenvalues, Eigenvectors

Elementary row operations, Rank of a matrix by echelon form, normal form, System of linear equations: Solving system of Homogeneous and Non-Homogeneous equations, Gauss elimination method, Inverse by Gauss-Jordan method, Gauss Seidel Iteration Methods**.** Eigenvalues, Eigenvectors and their properties (without proof).

UNIT II: Diagonalization of a matrix

Cayley-Hamilton Theorem (without proof), finding inverse and power of a matrix by Cayley-Hamilton Theorem, Diagonalization of a matrix, Quadratic forms and Nature of the Quadratic Forms, Reduction of Quadratic form to canonical forms by Orthogonal Transformation.

UNIT III: 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's theorems with remainders (without proof), Problems and applications on the above theorems. Taylor's and Maclaurin's series.

UNIT IV: Partial differentiation and Applications (Multi variable calculus)
Functions of several variables: Continuity and Differentiability, Partial derivatives, total derivatives, chain rule, Euler's homogeneous function 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 by the method of Lagrange multipliers.

UNIT V: 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).

TEXTBOOKS:

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

REFERENCE BOOKS:

1. B. V. Ramana, "Higher Engineering Mathematics", McGraw Hill Education, 2017.
2. Glyn James, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education, 2018.
3. H. K. Dass, "Advanced Engineering Mathematics", 22nd Edition, S. Chand Publishing, 2022.
4. R. K. Jain and S. R. K. Iyengar, "Advanced Engineering Mathematics", 5th Edition, Narosa Publications, New Delhi, 2016.

WEB RESOURCES

1. https://en.wikipedia.org/wiki/System_of_linear_equations
2. https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors
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
9. http://tutorial.math.lamar.edu/Classes/CalcIII/MultipleIntegralsIntro.aspx

B.Tech. (I Sem.) A2610003- COMPUTATIONAL THINKING & PROBLEM-SOLVING USING C

L	T	P	Cr.
3	0	0	3

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

Course Objectives: The main objectives of the course are to

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

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

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

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

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

UNIT	CONTENTS	Contact Hours
UNIT – 1	**Computational Thinking and Introduction to C** **Computational Thinking:** Decomposition, Pattern Recognition, Abstraction, Algorithm Design; representing solutions as algorithms, pseudocode, and flowcharts; Number Systems and Data Representation (integers, reals); Overview of Computer Organization and Program Execution. **Introduction to C:** Background, Structure of a C Program, Identifiers, Keywords, Variables, Constants, Basic Data Types, Input/Output statements; Expressions, Operators, type of operators- arithmetic, relational, logical, bitwise, assignment, conditional, increment, decrement, etc., unary, binary, ternary, Operator Precedence and Associativity, Type Conversion; Scope, Storage Classes, and Type Qualifiers.	12

UNIT – 2	**Selection and Decision-Making**: Simple if, Two-Way selection- if-else, nested-if, else-if ladder, and Multi-Way Selection- switch. **Repetition:** Counter- and Event-Controlled Loops: for, while, do-while, and nested loops; Break and continue, Programming Applications combining Selection and Repetition. **Arrays:** One-Dimensional and Two-Dimensional Arrays, Multidimensional Arrays, Array-based Problem Solving (searching, sorting, etc.).	9
UNIT – 3	**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. **Strings:** String Representation, Standard String-Handling Functions, Arrays of Strings, String-to-Data Conversion.	9
UNIT – 4	**Pointers:** Pointer Fundamentals, Pointer-to-Pointer, Pointer Arithmetic, Pointers and Arrays, Array of Pointers, types of pointers, Compatibility and L-value/R-value concepts. Dynamic Memory Allocation: malloc, calloc, realloc, free. **User-Defined Types:** typedef, Enumerated Types, Structures, Unions, and Nested Structures; Programming Applications using Structures and Arrays of Structures.	9
UNIT – 5	**Pre-processor**: Directives and Macros. Command Line Arguments. **File I/O:** Streams, Text vs. Binary Files, Standard Library File Functions, formatted and Character I/O, Sequential File Processing.	6
	Total	45

Text Books:

1. Behrouz A. Forouzan & Richard F. Gilberg, Computer Science: A Structured Programming Approach Using C, 3rd Edition, Cengage Learning, ISBN-13: 978-0-534-49132-1.
2. Brian W. Kernighan & Dennis M. Ritchie, The C Programming Language, 2nd Edition, Pearson Education / Prentice Hall India, 1988.

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.

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/

B.Tech. (I Sem.) **A2610004- DIGITAL LOGIC DESIGN**

L	T	P	Cr.
2	0	0	2

Pre-requisite: Basic Mathematics and Fundamentals of Computer Programming

Course Objectives

- Develop a strong foundation in number systems, binary codes, Boolean algebra, and logic gates used in digital systems.
- Enable students to apply Boolean algebra and Karnaugh map techniques for simplifying Boolean expressions and designing combinational logic circuits.
- Develop the ability to analyze and design combinational circuits using standard digital building blocks.
- Introduce the fundamental concepts of sequential logic, including latches and flip-flops, and their role in storing binary information.
- Build the ability to design and evaluate sequential circuits using shift registers and various types of counters.

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

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

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	-	-	-	-	-	-	-	-	-	M	-
CO2	H	H	H	M	M	-	-	-	-	-	-	-	H	-
CO3	H	H	H	M	M	-	-	-	-	-	-	-	H	-
CO4	H	H	M	M	M	-	-	-	-	-	-	-	H	-
CO5	H	H	H	M	M	-	-	-	-	-	M	-	H	-

UNIT	COURSE CONTENT
UNIT-I	**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
UNIT-II	**Combinational Logic Design I** Minimization using Karnaugh maps (up to 4 variables); Sum-of-Products and Product-of-Sums forms; design of half-adder, full-adder, half-subtractor, full-subtractor

UNIT-III	**Combinational Logic Design II:** Multiplexers and demultiplexers; encoders and decoders; magnitude comparators. Sequential Logic Fundamentals- Latches and Flip Flops, Difference between combinational and sequential circuits.
UNIT-IV	**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.
UNIT-V	**Registers, Counters, and Sequential Circuit Design** Shift registers (SISO, SIPO, PISO, PIPO); ripple (asynchronous) counters; synchronous counters; up/down and modulo-N counters

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 Open Courseware. (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/

B.Tech. (I Sem.) A2610005- DESIGN THINKING & INNOVATION

L	T	P	Cr.
0	1	3	2.5

Course Objectives

- 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*

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

Mapping of course outcomes with program outcomes

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

UNIT	COURSE CONTENT
UNIT-I	**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 Outcome** Students will be able to **understand users, identify their needs and translate observations into meaningful design opportunities**.
UNIT-II	**Problem Definition, Analysis & Opportunity Identification** From observations to insights; Data synthesis and affinity mapping; Identifying patterns and pain points; Problem statements; Point-of-View (POV) statements; "How Might We" (HMW) questions; Stakeholder mapping; 5 Whys and root-cause analysis; Problem prioritization; Need ~~O~~pportunity mapping; Problem validation; Design constraints and specifications; Social, environmental and sustainability considerations; Identifying opportunities for innovation **Practical Activities** • Affinity mapping of interview data • Develop three alternative problem statements Prepare POV and • HMW statements Stakeholder map • Root-cause analysis • Select and validate one problem/opportunity **Unit Outcome:** Students will be able to **analyze user research, identify root causes, frame well-defined problems and recognize opportunities for innovative solutions**
UNIT-III	**Creativity, Ideation & Concept Development** Creativity and innovation mindset; Divergent and convergent thinking; brainstorming and brainwriting; SCAMPER; Mind mapping; Reverse brainstorm~~ing~~g; Analogical thinking; Biomimicry; Lateral thinking; Forced connections; Morphological analysis; TRI basic principles; Individual and collaborative ideation; AI-assisted ideatio~~n~~: opportunities and limitations; Idea clustering and screening; Impact feasibility matrix; Concept selection; Design criteria and decision matrix **Practical Activities** • Generate 20–30 ideas for a selected problem SCAMPER exercise • Brainwriting exercise Biomimicry-based • ideation • AI-assisted idea generation and critical evaluation Prepare an idea-selection • matrix • Select the top 2–3 concepts **Unit Outcome** Students will be able to **generate diverse solution concepts using creative techniques and systematically select promising ideas based on defined criteria**.

UNIT-IV	**Prototyping, Experimentation & Validation** Concept development; Prototype mindset; Types of prototypes: Paper prototype, Physical prototype, Digital prototype, Storyboard, Roleplay, 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-V	**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 ; Cust~~om~~er 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**.

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. Atal Innovation Mission, NITI Aayog – Innovation and Design Thinking Resources. https://aim.gov.in/
2. Institution's Innovation Council (IIC), Ministry of Education, Government of India,

https://iic.mic.gov.in/

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

B.Tech. (I Sem.) A2610006- BASIC ELECTRICAL AND ELECTRONICS ENGINEERING

L	T	P	Cr.
2	0	0	2

Course Objectives: To provide students with fundamental knowledge of electrical circuits, AC circuits, electrical machines, semiconductor devices, measuring instruments, and operational amplifiers, and to develop their ability to analyze basic electrical and electronic systems using appropriate principles and techniques.

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

CO1: Understand fundamental electrical laws and network theorems. **(K2-Understand)**
CO2: Analyze AC circuits using phasors, impedance, power, power factor, and standard AC circuit parameters. **(K2-Understand)**
CO3: Understand the operating principles and basic characteristics of DC, BLDC, stepper, and servo motors. **(K2-Understand)**
CO4: Understand the operation and characteristics of semiconductor devices, rectifiers, BJT configurations, and basic electronic circuits. **(K2-Understand)**
CO5: Understand the working principles of electrical measuring instruments, digital meters, energy meters, and basic operational amplifier circuits **(K2-Understand)**

UNIT-I
DC Circuits and Network Theorems: Electrical circuit elements and terminology; Ohm's law; Kirchhoff's Current Law (KCL) and Voltage Law (KVL); series-parallel DC circuits; mesh and nodal analysis; Superposition theorem.

UNIT-II
AC Circuits: peak, average, RMS values, form factor and peak factor; Voltage and current relationship with phasor diagrams in R, L, and C series circuits only, concept of reactance and impedance, concept of power factor, active power, reactive power and apparent power.

UNIT-III
Electrical Machines: Principle of operation of DC Motor, BLDC motor, Stepper Motor and Servo motor.

UNIT-IV
Semiconductor Devices and Electronic Circuits: Operation of full-wave bridge rectifiers with and without filter; BJT: operation, input and output characteristics of CB & CE Configurations.

UNIT-V
Electrical Measuring Instruments and OP-Amps: Electrical measuring instruments: Operation of moving-coil and moving-iron instruments; digital multi meters, digital energy meters.
Operational Amplifiers: Introduction to op-amp, ideal operational amplifier, block diagram of op-amp, IC741 op-amp & its features, inverting, non-inverting amplifiers.

Learning Resources:
Textbooks

1. Ed Hughes, (2016), Electrical & Electronic Technology, 12th Edition, Pearson Education
2. Vincent Del Toro, (2015), Electrical Engineering Fundamentals, 2nd Edition, Pearson Education.

Reference Books

1. Mehta, V. K., & Mehta, R. (2008). Principles of electrical engineering and electronics. S. Chand Publishing.
2. Charles K. Alexander and Mathew N.O. Sadiku, (2017), Fundamentals of Electrical Circuits, (6th ed.), McGraw-Hill Education, India.

Web Resources

1. https://nptel.ac.in/courses/108105112
2. https://nptel.ac.in/courses/108101091
3. https://nptel.ac.in/courses/108105113
4. https://nptel.ac.in/courses/108105153

B.Tech. (I Sem.) A2610007- APPLIED CHEMISTRY IN EMERGING TECHNOLOGIES LAB

L	T	P	Cr.
0	0	3	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 characterization of nanomaterials and in safe laboratory practice, data analysis and technical reporting.

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

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** (Apply)

CO2: Determine electrochemical parameters including EMF, specific and equivalent conductance, and pH using conductometric, potentiometric and cell-based methods. **K3** (Apply)

CO3: Analyse the charge-discharge characteristics and performance of batteries and supercapacitors used in computing devices. **K4** (Analyze)

CO4: Synthesize and examine nanomaterials and evaluate their optical and adsorption properties for engineering applications. **K4** (Analyze)

CO5: Conduct experiments safely following laboratory protocols, and record, interpret and report data working individually and in teams. **K3** (Apply)

Mapping of course outcomes with program outcomes:

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

Scale: H - Strong | M - Moderate | L - Slight | - No Correlation.

LIST OF EXPERIMENTS

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

1. Verification of the photoelectric effect and estimation of Planck's constant.
2. Study of the emission spectrum of a hydrogen/mercury-vapour Lamp using a spectrometer.
3. Determination of the band-gap energy of a semiconductor sample from its optical absorption edge.
4. Determination of a strong acid against a strong base by using a conductometer.
5. Determination of the EMF of a Daniel cell and verification of the Nernst equation.
6. Determination of the charge-discharge characteristics of a lithium-ion battery/supercapacitor cell.
7. Determination of the specific and equivalent conductance of an electrolyte solution.
8. Assemble a simple Zn-carbon (or lead-acid) cell and measure its voltage and approximate capacity.
9. Construction and performance evaluation of a simple enzyme based electrochemical biosensor for glucose detection (demo)
10. Preparation and observation of colloidal gold/silver nanoparticles (size-dependent colour-change).
11. Determination of the viscosity of a polymer/nanomaterial-dispersion solution using an Ostwald viscometer.

12. Determination of the pH of an electronic-grade (deionized) water sample used in wafer/PCB cleaning.
13. Determination of strong acid against a strong base by using potentiometer.
14. Determination of conductivity of an electronic-grade (deionized) water sample used in wafer/PCB cleaning.
15. Redox titration by using potentiometer.
16. Determination of strength of an acid in Pb-Acid battery.
17. Adsorption capacity of a nanomaterial (e.g., activated carbon nanoparticles or nano-iron oxide) for removal of a dye or metal ion from water.
18. Study of galvanic corrosion using a zinc-iron electrochemical couple.

References:

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

L	T	P	Cr.
0	0	3	1.5

B.Tech. (I Sem.) **A2610008- COMPUTATIONAL THINKING & PROBLEM-SOLVING USING C LAB**

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*

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

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

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

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

List of Experiments

Exercise 1: Getting Started with

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

Exercise 2: Operators and Expressions

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

Exercise 3: Selection and Decision Making

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

Exercise 4: Loops and Series

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

Exercise 5: Nested Loops and Number Patterns

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

Exercise 6: One-Dimensional Arrays

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

Exercise 7: Matrices

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

Exercise 8: Strings:

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

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

- **Count the number of vowels, consonants, and words** in a given string

Exercise 9: Pointers Fundamentals

1. Write a C program to demonstrate the use of the & (address-of) and * (value-at-address) operators.
2. Write a C program to add two numbers using pointers.
3. Write a C program to swap two values using call by reference.

Exercise 10: Pointers and Dynamic Memory

1. Write a C program to dynamically allocate memory for an array using malloc, find the largest element and the sum of all elements in it, then repeat using calloc and note the difference between the two.
2. Write a C program to count the number of vowels in a string using a pointer.

Exercise 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.

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/

B.Tech. (I Sem.) **A2610009- IT AND ELECTRONICS WORKSHOP**

L	T	P	Cr.
0	0	3	1.5

Pre-requisites: NIL

Course Objectives (CO):

- Familiarize students with computer hardware, operating systems, peripheral devices and software installation.
- Develop practical skills in computer networking, internet technologies and collaborative platforms.
- To develop basic skills in MS Word, MS Excel, and MS PowerPoint for creating documents, working with data, and preparing presentations
- Develop hands-on skills in basic electronics, circuit assembly, soldering, and measurement, testing and troubleshooting.
- Develop an industry-oriented mini hardware prototype by integrating basic electronics, sensors, embedded software and appropriate documentation.

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

CO1: Assemble, configure and troubleshoot computer hardware, operating systems and software systems (K3).

CO2: Configure wired and wireless networks and use internet, cloud collaboration and related digital tools (K3).

CO3: Apply practical skills in Linux, MS Word, MS Excel, and MS PowerPoint to manage files, create documents, analyze data, and prepare presentations. (K3).

CO4: Apply practical skills to identify, test, assemble, solder, and measure basic electronic components and circuits. (K3).

CO5: Develop troubleshooting and problem-solving skills through the construction and demonstration of simple electronic circuits and mini projects (K5).

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

List of Experiments:

1. PC Hardware Component Identification and Testing
2. Desktop Computer Assembly
3. Windows/Linux Installation and Basic Troubleshooting
4. LAN Configuration and Network Cable Testing
5. Wi-Fi Router Configuration
6. Basic Linux Commands and File Management
7. Using MS Word, create a comprehensive document that integrates certificate creation, project abstract writing, and newsletter design while demonstrating all major formatting, layout, and editing features.
8. Using MS Excel, prepare a scheduler and GPA sheet applying formatting, formulas, charts, and multiple worksheets. Extend the project with data analysis using LOOKUP/ VLOOKUP,

sorting, logical operators, conditional formatting, and pivot tables.

9. Using MS PowerPoint, create an interactive presentation that demonstrates layouts, WordArt, multimedia insertion, hyperlinks, master layouts, views, backgrounds, and design templates.
10. Identification and Testing of Basic Electronic Components
11. Basic Electronic Circuit Assembly Using Breadboard
12. Soldering and Desoldering Practice
13. Measurement and Testing Using a Digital Multimeter
14. Construction and Troubleshooting of a Simple Electronic Circuit
15. Mini Project Development and Demonstration

Learning Resources:

Text Books:

1. V. K. Mehta and Rohit Mehta, Principles of Electronics, S. Chand & Company.
2. B. S. Raghuwanshi, A Course in Workshop Technology, Dhanpat Rai & Sons.
3. Vishnu P. Singh, Computer Hardware Course, Asian Publishers/Computech Publications.

Reference Books:

1. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dream tech, 2003
2. IIT Essentials PC Hardware and Software Labs and Study Guide, Patrick Regan– CISCO Press, Pearson Education, 3rd edition
3. Electronic Principles,Albert Malvino & David J. Bates,8th Edition,Components, breadboard circuits, measurements, troubleshooting.
4. Electronic Devices and Circuit Theory, Robert L. Boylestad & Louis Nashelsky,11th Edition,Basic electronic components, diode/transistor testing, simple circuits.

Online Resources:

1. **NPTEL – Computer Networks** – for networking fundamentals and practical concepts.
2. **NPTEL – Computer Networks and Internet Protocol, IIT Kharagpur** – for TCP/IP and Internet fundamentals.
3. **Ubuntu Official Documentation** – for Linux installation, file management, networking and system administration.
4. **NPTEL – Basic Electronics, IIT Bombay** – for basic electronic circuits and components.
5. **NPTEL – Basic Electronics and Lab, IIT Madras** – particularly useful for the practical electronics component.
6. **Tinkercad Circuits** – for virtual breadboard and circuit prototyping.
7. **Arduino Documentation** – for the mini-project/embedded-prototyping component.

B.Tech. (II Sem.)	A2620001-ENGLISH FOR TECHNICAL COMMUNICATION	L	T	P	Cr.
		3	0	0	3

Course Objectives:

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

Course Outcomes:

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

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: Evaluate the content, limitations and ethical aspects of AI tools in communication and language learning, and apply them to create concise, accurate essays/summaries/letters/paraphrases. (K5)

CO4: Examine discourse features and writing strategies to participate effectively in informal exchanges, short presentations on technical and general topics and to compile reports/E-mails. (K4)

CO5: Construct well-organized paragraphs, essays, critical reviews, and other professional documents suited to academic and workplace contexts. (K6)

Course Articulation Matrix:

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	2	–	–	1	2	–	–	1	3	1
CO2	1	1	1	–	1	3	2	2	3	3	2
CO3	2	3	2	2	3	1	1	3	2	3	3
CO4	1	2	1	1	1	2	1	1	3	3	3
CO5	1	2	1	1	1	2	1	1	3	3	3
1 = Slight (Low)			**2= Moderate (Medium)**				**3 = Substantial (High)**				

Unit 1 : Effective Communication: Exploring Channels and Methods **10 hrs**

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 **9 hrs**

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** **9 hrs**
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** **10 hrs**
Vocabulary: Business/ Profession/ Career/ Job-related Vocabulary.
Grammar: Sentence Synthesis, Simple, Compound and Complex Sentences.
Reading: Reading various texts for global understanding
Writing: Email Writing, Data Interpretation, Report Writing
Listening: Listening to debates and discussions.
Speaking: Power Point Presentations by Individuals and Teams.

Unit 5: **Ethical Leadership: Integrating Social Responsibility and Core Values** **10 hrs**
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). Next Gen 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 Black swan 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/

B.Tech. (II Sem.) A2620002-PHYSICS FOR ADVANCED COMPUTING

L	T	P	Cr.
3	0	0	3

Course Objective:

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

Course Outcomes: On completion of the course, the student should be able to

CO1: Apply the principles of interference, diffraction and polarization to analyze optical systems and engineering applications. **Apply (K3)**

CO2: Analyze the operation of various laser systems as well as the working principle of optical fibers in relation to communication applications. **Analyze (K4)**

CO3: Analyze the operation of various laser systems as well as the working principle of optical fibers in relation to communication applications. **Interpret (K3)**

CO4: Analyze semiconductor characteristics and magnetic material behavior for engineering and electronic applications. **Analyze (K4)**

CO5: Apply the concepts of low-temperature physics and superconductivity in modern technological applications and intelligent systems. **Apply (K3)**

Mapping of course outcomes with program outcomes:

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	H	M	–	–	M	–	–	–	–	L	M
CO2	H	M	M	L	H	–	–	–	L	L	M
CO3	H	H	M	M	H	–	–	–	L	L	M
CO4	M	H	M	M	M	L	–	–	L	L	M
CO 5	H	M	M	M	L	L	–	–	–	–	–

Correlation levels: H = High (3), M = Moderate (2), L = Low (1), – = No correlation (0)

UNIT – I: APPLIED WAVE OPTICS **10 Hrs**

Interference: Superposition Principle – Coherence – Types of Interference – Thin-film interference (reflection geometry) – Newton's rings – Determination of wavelength and refractive index.

Diffraction: 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–II: QUANTUM PHOTONICS **9 Hrs**

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.

UNIT – III: FREE ELECTRON THEORY OF METALS & QUANTUM MECHANICS 10 Hrs

Classical free electron theory – merits and demerits – Quantum free electron theory – equation for electrical conductivity – Fermi-Dirac statistics – Fermi level and variation with temperature.

Quantum Mechanics: de Broglie concept of matter waves – Heisenberg's uncertainty principle – Wave function – Schrödinger's time dependent and time-independent wave equations – Particle in a one-dimensional potential well – Quantum tunneling (basic idea).

UNIT – IV: SEMICONDUCTORS & MAGNETIC MATERIALS 10 Hrs

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.

UNIT – V: LOW TEMPERATURE PHYSICS & SUPERCONDUCTIVITY 9 Hrs

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.

Learning Resources:

Textbooks:

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).
3. B. K Pandey and S. Chaturvedi, Engineering Physics, 2nd Edition, CENGAGE learning (2021).

Reference Books:

1. Fundamentals of Physics by David Halliday, Robert Resnick, and Jearl Walker, John Wiley & Sons publications, 12th Edition (2021).
2. Solid State Physics by S. O. Pillai, New Age International Publishers, 9th edition (2023).
3. Concepts of Modern Physics by Arthur Beiser, Shobit Mahajan & S. Rai Choudhury MC Graw Hill, 8th Edition (2026).
4. Nielsen & Chuang, Quantum Computation & Quantum Information, Cambridge University Press, 10th Edition, (2010).
5. D. J. Griffiths, Introduction to Electrodynamics, Cambridge University Press, 5th Edition (2023).

Web Resources

1. NPTEL. National Programme on Technology Enhanced Learning. https://nptel.ac.in
2. MIT Open Course Ware, Physics courses. https://ocw.mit.edu
3. Georgia State University, Hyper Physics. http://hyperphysics.phy-astr.gsu.edu
4. PhET Interactive Simulations, University of Colorado Boulder. https://phet.colorado.edu
5. American Concrete Institute, (2021). ACI 228.2R: Nondestructive test methods for evaluation of concrete in structures. https://www.concrete.org

B.Tech. (II Sem.) A2620003- DIFFERENTIAL EQUATIONS & VECTOR CALCULUS

L	T	P	Cr.
3	0	0	3

Course Objectives: It is aimed

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

Course Outcomes: Upon completion of the course, the student will be able to:

CO1	**Create** a Mathematical model for simple physical problems. (K6)
CO2	**Apply** second order differential equations to analyse L - C - R circuits. (K3)
CO3	**Formulate** and **solve** the partial differential equations that model physical processes. (K6)
CO4	**Apply** knowledge of Gradient, divergence, Curl and interpret the physical meaning of different operators. (K3)
CO5	**Compute** the work done against a field, circulation and flux using vector calculus methods. (K3)

K1- Remembering, K2- Understanding, K3-Applying, K4- Analyzing, K5- Evaluating, K6- Creating

Course Articulation Matrix

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

H – High, M - Medium, L - Low

UNIT I: Differential equations of first order and first degree

Exact equations and equations reducible to exact form (integrating factor), Linear differential equations – Bernoulli's equations. Applications: Orthogonal Trajectories in Cartesian Coordinates, Newton's Law of cooling – Law of natural growth and decay- Electrical circuits.

UNIT II: Linear differential equations of higher order (Constant Coefficients)

Definitions of linear and non – linear equations, homogenous and non-homogenous equations, complementary function, particular integral, general solution, Method of variation of parameters - Wronskian. Simultaneous linear equations, Applications to L-C-R Circuit problems.

UNIT III: Partial Differential Equations

Introduction and formation of Partial Differential Equations by elimination of arbitrary constants and arbitrary functions, solutions of first order linear equations using Lagrange's method. Homogeneous Linear Partial differential equations with constant coefficients.

UNIT IV: Vector differentiation

Scalar and vector point functions, vector operator Del, Gradient and applications, Directional derivative, Divergence - Solenoidal and Curl - Irrotational Field - Scalar Potential, Vector identities.

UNIT V: Vector integration

Line integral-circulation-work done by the force, surface integral-flux, Green's theorem in a plane (without proof), Stoke's theorem (without proof), volume integral, Divergence theorem (without proof) and their applications.

TEXTBOOKS:

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

REFERENCE BOOKS:

1. B. V. Ramana, "Higher Engineering Mathematics", McGraw Hill Education, 2017.
2. Glyn James, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education, 2018.
3. H. K. Dass, "Advanced Engineering Mathematics", 22nd Edition, S. Chand Publishing, 2022.

WEB RESOURCES

1. https://mathworld.wolfram.com/First-OrderOrdinaryDifferentialEquation.html
2. https://en.wikipedia.org/wiki/Differential_equation
3. https://en.wikipedia.org/wiki/Partial_differential_equation
4. https://en.wikipedia.org/wiki/Vector_calculus
5. https://en.wikipedia.org/wiki/Vector_calculus

B.Tech. (II Sem.) **A2620004- DATA STRUCTURES USING C**

L	T	P	Cr.
3	0	0	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
CO1	Explain fundamental data structure concepts, abstract data types, classifications, and analyze the time and space complexity of algorithms. (K2-Understand)
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-Apply)
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-Analyze)
CO4	Implement and apply tree data structures, including binary trees, binary search trees, balanced binary search trees, and heaps, using appropriate traversal and manipulation techniques. (K3-Apply)
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-Apply)

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

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

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

Text 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.

Reference Books:

1. Deshpande, P. S., & Kakde, O. G. C and data structures. Dreamtech Press.
2. 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

B.Tech. (II Sem.) **A2620005- PYTHON PROGRAMMING**

L	T	P	Cr.
2	0	0	2

Pre-requisite: Problem Solving and Programming Using C.

Course Objectives: The main objectives of the course are to

- Understand the fundamental concepts and programming features of Python, including variables, data types, operators, input/output, and control structures.
- Make use of Python data structures such as strings, lists, tuples, dictionaries, and comprehensions for handling different types of data.
- Develop familiarity with functions, recursion, lambda expressions, modules, libraries, and basic programming efficiency in Python.
- Understand object-oriented programming concepts in Python, including classes, objects, encapsulation, inheritance, polymorphism, and composition.
- Make use of file handling, exception handling, NumPy, pandas, and Matplotlib for developing basic data processing and visualization applications.

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

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Explain the fundamental concepts of Python programming, including variables, data types, operators, input/output, conditional statements, and loops.	K2
CO2	Apply Python data structures such as strings, lists, tuples, dictionaries, and comprehensions to store and manipulate data	K3
CO3	Apply functions, recursion, lambda expressions, modules, and libraries to develop modular Python programs.	K3
CO4	Analyze object-oriented programming concepts such as classes, objects, encapsulation, inheritance, polymorphism, and composition in Python.	K4
CO5	Apply file handling, exception handling, NumPy, pandas, and Matplotlib for basic data processing and visualization.	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	M	-	H	-	-	-	-	-	-	M	M	L
CO2	H	H	H	L	H	-	-	-	-	-	-	H	M	M
CO3	H	H	H	M	H	-	-	-	-	-	M	H	H	M
CO4	H	H	H	M	H	-	-	-	-	-	M	H	H	M
CO5	H	H	H	H	H	-	-	-	-	M	M	H	M	H

UNIT	CONTENTS
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.
UNIT – 2	**DATA STRUCTURES IN PYTHON** **Sequences and Collections:** Difference between a sequence and a collection, types of sequences and collections in python. **Strings**: Pattern matching and abstraction of text as a sequence; string operations and methods. **Lists:** Abstraction of ordered, mutable collections; list operations and methods **Tuples:** Abstraction of fixed (tuples) data, tuple operations and methods **Dictionaries:** Key-based (dictionaries) data; common operations, methods and use-case selection. **Sets:** Unordered representation of data, set operations and methods **Comprehensions:** Significance of comprehension, List, Set and Dictionary Comprehensions.
UNIT – 3	**FUNCTIONS, MODULES AND RECURSION** **Functions:** Need for functions, Function definition, Decomposition of a program into independent, reusable sub-problems; Parameters and arguments , Return values and Scope. **Modules and Libraries:** Importing modules, Built-in modules, Creating user-defined modules, Packages and Built-in libraries. **Recursion:** Concept of recursion, Base condition, Recurrence relation, Algorithmic thinking applied to self-similar sub-problems. Algorithmic Efficiency: Informal comparison of iterative vs. recursive approaches and their time trade-offs. **Lambda Expressions and Functional Tools:** lambda, map, filter, and reduce, concise transformations and their use cases. Responsible AI-assisted coding practices.
UNIT – 4	**OBJECT ORIENTED PROGRAMMING** **Introduction to OOP**: Features of object-oriented programming, encapsulation, abstraction, inheritance, polymorphism. **Classes and Objects:** Class definition, the __init__ method, class variables and instance variables, public, private and protected data members, calling methods from other methods, built-in class attributes. **Inheritance:** Inheriting classes, types of inheritance, method overriding. **Polymorphism and Operator Overloading:** Polymorphism through method overriding; operator overloading. **Composition:** Composition (has-a) vs. inheritance (is-a) relationships.
UNIT – 5	**EXCEPTIONS, FILE HANDLING AND DATA ANALYSIS** **Exception Handling**: try-except-finally; abstraction of error cases into predictable, manageable patterns; user-defined exceptions. **File Handling:** Reading and writing files; decomposition of a data workflow into read/process/write stages. **Data Analysis and Visualization**: Introduction to NumPy, pandas and Matplotlib basics; pattern recognition in real datasets.

Text Books:

1. Downey, A. B. (2015). Think Python: How to think like a computer scientist (2nd ed.). O'Reilly Media.
2. J. V. Guttag, Introduction to Computation and Programming Using Python: With Application to Computational Modeling and Understanding Data, 3rd Edition, MIT Press, 2021.

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. 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/

B.Tech. (II Sem.) A2620006-ENGLISH FOR TECHNICAL COMMUNICATION AND MOODLES LAB

L	T	P	Cr.
0	0	3	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 speaking 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.

Course Articulation Matrix:

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
L=Slight (Low)				M= Moderate (Medium)					H = Substantial (High)		

List of Experiments

1. **Phonetics & Accent Neutralisation**

- Sound Lab Drills — Students record minimal pairs (/p/–/b/, /t/–/d/) and compare wave forms 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. **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.

6. **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.

7. **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.

8. **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).

9. **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.

10. **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

Moodle Learning Management System: Digital learning and 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, Culture Wizard, 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
- Re-submission and revision

4. **Academic honesty**

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

@ @ @

B.Tech. (II Sem.) **A2620007- DATA STRUCTURES USING C LAB**

L	T	P	Cr.
0	0	3	1.5

Pre-requisite: Programming for Problem Solving using C

Course Objectives: The main objectives of the course are to

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

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

CO	Course Objective Description	Knowledge Level (K)#
CO1	Write and debug C programs using pointers and dynamic memory allocation to implement array-based searching and sorting techniques.	K3
CO2	Build and manipulate singly, doubly and circular linked lists, and use them to solve problems such as polynomial arithmetic and duplicate removal.	K3
CO3	Implement stack-based and queue-based programs for expression conversion/evaluation, parenthesis matching and recursion-related problems.	K3
CO4	Construct binary search trees and implement all traversal techniques to solve hierarchical-data problems.	K3
CO5	Design and test hash-table programs using chaining and open-addressing collision-resolution techniques.	K4

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

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

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

List of Experiments

Exercise 1: Array-Based Techniques, searching

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

Exercise 2: Sorting

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

Exercise 3: Singly Linked List

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

Exercise 4: Singly Linked List Applications

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

Exercise 5: Doubly and Circular Linked Lists

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

Exercise 6: Stack Implementation

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

Exercise 7: Stack Applications

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

Exercise 8: Recursion and Stack Simulation

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

Exercise 9: Queue and Circular Queue

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

Exercise 10: Binary Trees

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

Exercise 11: Binary Search Trees

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

Exercise 12: Tree Traversals and Problems

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

Exercise 13: Hashing

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

Web References:

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

B.Tech. (II Sem.) A2620008- AI TOOLS & APPLICATIONS

L	T	P	Cr.
0	1	3	2.5

Course Outline:

- This course introduces first-year engineering students to the practical use of Artificial Intelligence tools across problem-solving, data analysis, content creation, and coding.
- Students gain hands-on experience with generative AI, prompt engineering, AI-assisted data visualization, and productivity/coding copilots through weekly lab practice.
- The course emphasizes responsible and ethical AI use alongside technical competence and serves as a foundation for AI-assisted work in higher-semester courses across all branches.
- By the end, students will be able to independently select, apply, and evaluate AI tools for academic and engineering tasks.

Course Objectives: The main objectives of the course are to

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

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

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Recall the categories, capabilities, and limitations of major AI tools (text, image, code, data).	K1
CO2	Explain the principles of prompt engineering and how generative AI models process instructions.	K2
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

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

<table>
<tr><th>UNIT</th><th>CONTENTS</th><th>Contact Hours</th></tr>
<tr><td>UNIT – 1</td><td>
Introduction to AI and the AI Tools Landscape
What is AI, Machine Learning, and Generative AI — basic definitions and how they relate to one another.
Core categories of AI tools: conversational/text, code assistants, data-analysis tools, image-generation tools, and writing/productivity tools.
Large language models process and generate responses : conceptual, non-mathematical overview.
Criteria for selecting the right AI tool for a given task — accuracy needs, cost, data sensitivity, and output format.
Setting up accounts and workspaces: understanding free-tier versus paid-tier capabilities and limitations.
Widely Used AI Tools by Application:
<table>
<tr><th>Category</th><th>Example Tools</th><th>Typical Application</th></tr>
<tr><td>Conversational / General-Purpose AI</td><td>ChatGPT, Claude, Gemini, Perplexity</td><td>Writing, research, coding help, file/data analysis, cited web search</td></tr>
<tr><td>AI Coding Assistants</td><td>GitHub Copilot, Cursor, Claude Code</td><td>In-IDE code completion, AI-native editing, multi-file agentic coding</td></tr>
<tr><td>Data Analysis Tools</td><td>ChatGPT (Advanced Data Analysis), Microsoft Copilot in Excel, Julius AI</td><td>Upload a dataset, generate statistics/charts, spreadsheet formulas via natural language</td></tr>
<tr><td>Image Generation</td><td>Midjourney, DALL·E 3, Stable Diffusion / Flux</td><td>Illustrations, concept art, design mockups</td></tr>
<tr><td>Writing & Productivity</td><td>Grammarly, Gamma, Fathom, Notion AI</td><td>Grammar/style checking, slide generation, meeting summaries, task automation</td></tr>
<tr><td>Audio, Video & Presentation (optional)</td><td>ElevenLabs, Runway/Sora, Synthesia</td><td>Text-to-speech, AI video generation, script-to-video presentations</td></tr>
</table>
Unit Outcome(s):
1. Identify the categories of AI tools available and select an appropriate tool for a given academic or engineering task.
Teacher Practical Examples (for classroom demonstration):
1. Live comparison of the same general engineering query (e.g., "explain the engineering design process") answered by two different AI chatbots, highlighting differences in response style and accuracy.
2. Walkthrough of an AI tool's interface — settings, conversation history, file upload, and model selection — using a common platform.
Student Exercise Problems (for practice):
1. Create accounts on two AI platforms and compare their responses to a simple engineering question (e.g., "explain the difference between analysis and design").
2. Prepare a one-page comparison chart of five AI tools by category and primary use case.
</td><td>9</td></tr>
</table>

	3. Write a short reflection identifying one task in your daily academic life where an AI tool could help.	
UNIT – 2	**Prompt Engineering and Generative AI Tools** **Anatomy of a good prompt** — context, clear instruction, desired format, and constraints. **Types of prompt Techniques:** Zero-shot, few-shot, and chain-of-thought prompting techniques and when to use each. **Role-based and context-setting prompts** (e.g., asking the AI to respond as a reviewer, tutor, or domain expert). **Iterative prompt refinement** — diagnosing a poor AI response and rewriting the prompt to fix it. **Using generative AI** ----for brainstorming, summarizing, and drafting technical content. **Unit Outcome(s):** 1. Construct clear, well-structured prompts using zero-shot, few-shot, and chain-of-thought techniques to obtain reliable AI output. **Teacher Practical Examples (for classroom demonstration):** 1. Refine a vague prompt (e.g., "explain project scheduling") into a precise one step-by-step, showing output improvement at each stage. 2. Demonstrate few-shot prompting by providing examples to guide the AI toward a specific output format (e.g., a formatted comparison table). **Student Exercise Problems (for practice):** 1. Write three versions of a prompt (basic, detailed, role-based) for the same task and compare outputs. 2. Use an AI tool to summarize a one-page technical article and verify the summary against the original. 3. Draft a chain-of-thought prompt to solve a multi-step numerical/logical problem (e.g., a unit-conversion or estimation problem) and check the reasoning.	9
UNIT – 3	**AI Tools for Data Analysis and Visualization** AI-assisted spreadsheet and dataset analysis using natural-language commands. Natural-language querying of datasets to extract summary statistics and insights. AI tools for generating charts, graphs, and visual summaries from tabular data. Interpreting AI-generated statistical summaries correctly and recognizing their limitations. Identifying and correcting errors or misinterpretations in AI-generated data analysis. **Unit Outcome(s):** 1. Use AI tools to clean, summarize, and visualize a dataset via natural-language commands, and identify errors in AI-generated analysis. **Teacher Practical Examples (for classroom demonstration):** 1. Upload a small dataset (e.g., class attendance or survey data) to an AI tool and generate a summary-statistics report and chart using natural-language commands. 2. Show an example where an AI tool misinterprets data (e.g., treats a numeric code as text), and demonstrate how to verify and correct it. **Student Exercise Problems (for practice):** 1. Use an AI tool to clean and summarize a small sample dataset (e.g., class	9

	attendance or marks data). 2. Generate three different chart types from the same dataset using AI prompts and justify which is most appropriate. 3. Cross-check one AI-generated statistical claim manually and report any discrepancy found.	
UNIT – 4	**AI Tools for Coding, Writing, and Productivity** AI coding assistants — code generation, explanation, and debugging support for simple programs. AI writing assistants for reports, emails, and technical documentation. AI tools for scheduling, note-taking, and meeting/lecture summarization. AI-assisted presentation and slide-outline generation from a topic prompt. Integrating AI tools into an efficient day-to-day academic and engineering. **Unit Outcome(s):** 1. Apply AI coding assistants and writing tools to complete and refine a simple technical task. 2. Integrate multiple AI tools into a coherent workflow. **Teacher Practical Examples (for classroom demonstration):** 1. Use an AI coding assistant to generate a simple program (e.g., a temperature converter), then intentionally introduce a bug and have the AI debug it. 2. Generate a presentation outline and speaker notes from a topic prompt using an AI writing/productivity tool. **Student Exercise Problems (for practice):** 1. Use an AI coding tool to write and debug a simple program for a given problem statement. 2. Draft a formal email or short report using an AI writing assistant, then manually edit it for accuracy and tone. 3. Generate a slide outline for a 5-minute seminar topic using an AI tool and refine it.	9
UNIT – 5	**Ethics, Reliability, and Emerging Trends** Bias and hallucination in AI outputs, and techniques for detecting and fact-checking them. Academic integrity and appropriate citation of AI assistance in coursework and submissions. Data privacy and security considerations, including obligations under India's Digital Personal Data Protection Act, 2023 (DPDP Act), when uploading personal or institutional data. Copyright and intellectual property considerations in AI-generated text and image content. Emerging trends — multimodal AI and AI agents — and their implications for future engineering practice. **Unit Outcome(s):** 1. Evaluate AI-generated output for hallucination, bias, and copyright concerns, and apply ethical and data-privacy principles to AI tool use. **Teacher Practical Examples (for classroom demonstration):** 1. Show a documented case of AI hallucination (a fabricated fact or citation) and demonstrate how to fact-check it. 2. Walk through an example of properly citing AI assistance in an academic submission per institutional guidelines. 3. Walk through a worked example of checking whether AI-generated text or image output raises a copyright/IP concern (e.g., close paraphrasing of a source).	9

	Student Exercise Problems (for practice): 1. Identify and document a hallucination or bias instance produced by an AI tool on a factual query. 2. Draft an "AI usage declaration" for a hypothetical assignment, citing tools and extent of use appropriately. 3. Review the data-handling/privacy policy of one AI tool used in this course and identify one DPDP Act, 2023 compliance consideration relevant to student data.	
	Total	**45**

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

Review Questions

1. Explain the difference between AI, Machine Learning, and Generative AI with a suitable example.
2. Describe the anatomy of a good prompt and illustrate zero-shot versus few-shot prompting with an example.
3. Discuss two ways AI tools support data analysis and visualization, with an example of each.
4. What is meant by hallucination in AI-generated output? Explain how it can be detected and verified.
5. Discuss the ethical, academic-integrity, and data-privacy considerations relevant to using AI tools for coursework, referencing the DPDP Act, 2023.

Practical Applications

1. Using AI tools to accelerate literature review and topic understanding for mini-projects and seminars.
2. Applying AI-assisted data analysis to quickly clean, summarize, and visualize experimental or survey data.
3. Using AI coding assistants to write, explain, and debug simple programs across engineering assignments.
4. Leveraging AI writing assistants to draft clear reports, emails, and presentations for academic and professional communication.
5. Applying responsible AI-usage practices — verification, citation, and privacy awareness — in everyday academic work.

B.Tech. (II Sem.) **A2620009-PYTHON PROGRAMMING LAB**

L	T	P	Cr.
0	0	2	1

Pre-requisite: Concurrent with Python Programming

Course Objectives: The main objectives of the course are to

- Understand the basic features of Python programming and control structures through hands-on problem solving.
- Make use of Python data structures for storing, organizing, and manipulating different types of data.
- Develop familiarity with functions, recursion, modules, lambda expressions, and functional programming tools for modular programming.
- Understand object-oriented programming concepts for modelling real-world problems in Python.
- Make use of exception handling, file processing, and basic data analysis and visualization using Python libraries.

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

CO	Course Outcome Description	Knowledge Level (K)#
CO1	Apply Python programming constructs, including variables, operators, input/output, conditional statements, and loops, to solve computational problems.	K3
CO2	Implement Python programs using strings, lists, tuples, dictionaries, and comprehensions to represent and manipulate data.	K3
CO3	Develop modular Python programs using functions, recursion, lambda expressions, modules, and libraries.	K3
CO4	Analyze object-oriented Python programs using classes, objects, encapsulation, inheritance, polymorphism, and composition.	K4
CO5	Apply exception handling, file processing, NumPy, pandas, and Matplotlib for practical data processing and visualization tasks.	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	M	–	H	–	–	–	–	–	–	H	M	L
CO2	H	H	H	L	H	–	–	–	–	–	–	H	M	M
CO3	H	H	H	M	H	–	–	–	–	–	M	H	H	M
CO4	H	H	H	M	H	–	–	–	–	–	M	H	H	M
CO5	H	H	L	H	H	L	L	L	L	M	M	H	M	H

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 personalized 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: Conditionals 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: Lists and List comprehensions 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 09: Recursion vs Iteration

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

Exercise 10: Exception Handling

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

Exercise 11: 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

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

B.Tech. (II Sem.) A2620010-PHYSICS FOR ADVANCED COMPUTING LAB

L	T	P	Cr.
0	0	3	1.5Z

Course Objective:

1. To develop experimental understanding of fundamental principles of optics, semiconductor physics, quantum physics, magnetism, electromagnetism, and dielectric materials relevant to advanced computing technologies.
2. To experimentally determine important optical, electrical, electronic, magnetic, dielectric, and material parameters using appropriate measurement techniques and instrumentation.
3. To develop skills in experimental data acquisition, graphical analysis, interpretation of characteristics, and application of physical principles to understand the behavior of materials and devices used in modern electronic and computing systems.

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

CO1: Apply the principles **of** interference, diffraction, and polarization to experimentally determine optical parameters such as radius of curvature, wavelength, grating element, and Brewster angle. **K3 (Apply)**

CO2: Analyze the electrical and electronic characteristics of semiconductor materials and devices to determine parameters such as temperature coefficient, energy gap, and Hall coefficient using appropriate experimental techniques. **K4 (Analyze)**

CO3: Apply the principles of quantum physics and photoelectric effect to experimentally estimate Planck's constant and relate experimental observations to fundamental quantum concepts. **K3 (Apply)**

CO4: Analyze the magnetic properties and electromagnetic field characteristics of materials and current-carrying systems by studying B–H characteristics and the magnetic field distribution along the axis of a circular coil. **K4 (Analyze)**

CO5: Analyze the dielectric and temperature-dependent electrical properties of materials through charging–discharging and thermistor characteristics, and interpret the corresponding experimental behavior. **K4 (Analyze)**

CO–PO Articulation Matrix:

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	H	H	–	H	M	–	–	–	L	M	–
CO2	H	H	–	H	H	–	–	–	L	M	–
CO3	H	H	–	H	M	–	–	–	L	M	–
CO4	H	H	–	H	H	–	–	–	L	M	–
CO5	H	H	–	H	H	–	–	–	L	M	–

Correlation Levels: H – High correlation; **M** – Moderate correlation; **L** – Low correlation

LIST OF EXPERIMENTS

1. Determination of the radius of curvature of a given Plano convex lens by forming Newton's rings.
2. Determination of the wavelength of a Laser and a grating element by using a grating.
3. Determination of temperature coefficients of a thermistor.
4. Determination of the energy gap of a semiconductor using a p-n junction diode.
5. Estimation of Planck's constant using the photoelectric effect.
6. Study the variation of B versus H by magnetizing the magnetic material (B-H curve).
7. Study the variation of magnetic field along the axis of a current carrying circular coil by Stewart & Gee's method.
8. Determination of the dielectric constant of a given capacitor by the charging and discharging method.
9. Determination of Hall voltage and Hall coefficient of a given semiconductor using Hall Effect.

10. Verification of Brewster's law.
11. Determination of wavelengths of different spectral lines in the mercury spectrum using a diffraction grating in normal incidence configuration.
12. Determination of the resolving power of a telescope.
13. Determination of the Resistivity of a semiconductor by the four probe method.
14. Determination of the thickness of a thin object by the wedge method.
15. Determination of Acceleration due to gravity and Radius of gyration using a Compound pendulum.
16. Determination of magnetic susceptibility by Kundt's tube method.
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

B.Tech. (II Sem.) A2620011-ENVIRONMENTAL SCIENCE

L	T	P	Cr.
2	0	0	0

Course Objectives:

- To understand the fundamental structures and interactions of natural ecosystems biogeochemical cycles, and Earth's natural resources
- To evaluate the causes, impacts, and mitigation strategies of various forms of environmental pollution (air, water, soil, noise)
- To analyze current global environmental issues, climate change mechanisms, and principles of sustainable development and green engineering.
- To apply social, legal, and ethical frameworks to environmental management, environmental impact assessments (EIA), and corporate social responsibility.
- To propose sustainable engineering concepts and analytical solutions for contemporary environmental problems through case-based design approaches.

Course Outcomes: On completion of the course, the student should be able to

CO1: Assess natural resource distribution, ecological balance mechanisms, and biodiversity index parameters.

CO2: Analyze pollution control options and waste handling strategies for air, water, and solid waste systems based on regulatory guidelines.

CO3: Evaluate carbon footprints, evaluate climate change mitigation policies, and integrate life-cycle thinking into industrial plans.

CO4: Understand environmental legislation, outline introductory Environmental Impact Assessments (EIAs), and practice professional environmental ethics.

CO5: Implement green building concepts, renewable energy applications, and circular economy practices conceptually in professional environments.

Course articulation matrix

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	M	M	–	–	–	L	H	–	–	–	–	–			
CO2	M	H	–	–	–	H	H	L	–	–	–	–			
CO3	M	H	–	–	–	H	H	L	–	–	–	–			
CO4	L	M	–	–	–	H	H	H	–	–	–	–			
CO5	M	M	–	–	–	H	H	L	–	–	–	–			

UNIT I: Ecosystems, Natural Resources, and Biodiversity

Concept, structure, and functions of an ecosystem - Producers, consumers, decomposers; **Energy flow in ecosystems:** Food chains, food webs, ecological pyramids; **Biogeochemical cycles:** Carbon, Nitrogen, Phosphorus, and Water cycles; **Natural resources:** Renewable and non-renewable resources - Forest, Water, Mineral, Energy and Food resources and associated exploitation challenges; **Biodiversity:** Levels - genetic, species, ecosystem, values, hot-spots, threats, and conservation methods (In-situ and Ex-situ); **Case Studies & Activities.**

UNIT II: Environmental Pollution and Control Technologies

Air Pollution: Sources, primary/secondary pollutants, ambient air quality standards, control equipment principles - Electrostatic Precipitators, Baghouses, Scrubbers; **Water Pollution:** Characteristics of wastewater, parameters - BOD, COD, TDS, municipal wastewater treatment principles - Primary, Secondary, Tertiary treatment; **Soil Pollution:** Industrial waste, agricultural runoff, solid waste management Municipal Solid Waste - generation, segregation, incineration, composting, landfilling; **Noise and Thermal Pollution:** Sources, measurement metrics (dB, dBA), physiological impacts, and mitigation techniques; **Case Studies & Activities.**

UNIT III: Global Environmental Issues and Climate Change

The Earth's climate system, enhanced greenhouse effect, and global warming potential (GWP); **Ozone layer depletion:** CFCs, catalytic destruction cycles, and the Montreal Protocol; **Acid rain formation** - atmospheric chemistry impacts, and regional degradation; **International Conventions and Policies**: UNFCCC, Paris Agreement, Kyoto Protocol, and Carbon Trading mechanisms; **Concepts of Life Cycle Assessment (LCA)**: Carbon Footprint analysis, and Net-Zero carbon engineering; **Case Studies & Activities.**

UNIT IV: Environmental Ethics, Laws, and Sustainable Development

Sustainable Development Goals (SDGs) set by the United Nations: Targets and engineering intersections; **Environmental Legislation**: Overview of Water (Prevention and Control of Pollution) Act, Air Act, Environment (Protection) Act, and Wildlife Protection Ac; **Environmental Impact Assessment (EIA)**: Process, scoping, public consultation, clearance lifecycle, and Environmental Management Plan (EMP); **Environmental Ethics:** Corporate Social Responsibility (CSR); **Case Studies & Activities.**

UNIT V: Green Technology and Sustainable Engineering

Renewable Energy Technologies: Solar photovoltaic, wind energy, biomass conversion, and hydrogen fuel cells; **Green Building Concepts:** Leadership in Energy and Environmental Design (LEED), IGBC ratings, energy-efficient architecture, and waste-water recycling loops; **Circular Economy:** Upcycling, industrial symbiosis, waste-to-energy recovery systems, and sustainable supply chains; **Case Studies & Activities and Field Activity(Plantation, Species identification, Study of ecosystems etc)**

Textbook:

1. **Rajagopalan, R.** (2023). *Environmental Studies: From Crisis to Cure*, 4th Edition. Oxford University Press, New Delhi. ISBN: 9789354978944.
2. **Masters, G. M., & Ela, W. P.** (2013). *Introduction to Environmental Engineering and Science* (3rd ed.). Pearson.
3. **Bharucha, E.** (2021). *Textbook of Environmental Studies for Undergraduate Courses*. Universities Press.
4. **Kaushik, Anubha and Kaushik,** C. P. (2021). *Perspectives in Environmental Studies*, 7th Edition. New Age International Pvt. Ltd., New Delhi.

Reference Books:

1. **De, A. K.** (2010). *Environmental Chemistry* (7th ed.). New Age International Publishers.
2. **Nebel, B.J., & Wright, R.T.** (2008). *Environmental Science: Toward a Sustainable Future.* Pearson.
3. KVSG Murali Krishna, The Book of Environmental Studies, 2/e, VGS Publishers, 2011.
4. **UN Environment Programme (UNEP):** Official publications and Sustainable Development Goals resource repositories [unep.org].
5. **Coursera / edX Online Modules:** Introduction to Sustainability or Circular Economy and Sustainable Technologies (DelftX / LundX).
6. NPTEL, Environmental Science, Prof. Sudha Goel and Prof. Shamik Chowdhury, IIT Kharagpur,NPTEL–SWAYAM. https://onlinecourses.nptel.ac.in/noc26_hs120/preview?utm_source=chatgpt.com