

24MA101 Mathematics – I (Linear Algebra, Series and Calculus)			
Course Category:	Basic Sciences	Credits:	4
Course Type:	Theory	Lecture-Tutorial-Practice:	3-0-2
Pre-requisites:	10+2 Mathematics	Continuous Assessment:	40
		Summative Assessment:	60
		Total Marks:	100

Course Description: An overview of the fundamental concepts of linear algebra, infinite series, Differential calculus, multiple integrals and vector calculus, with a focus on the applications in solving engineering problems.

Course Objectives:

- Introduce techniques for solving systems of linear equations, determining Eigen values and eigenvectors, and performing matrix diagonalization.
- Explain methods to analyze the convergence and divergence of an infinite series and to expand functions using Taylor's or Maclaurin's series
- Familiarize differentiation rules and theorems to solve problems related to rates of change and optimization
- Explain the concept of double and triple integrals to calculate areas and volumes for two-dimensional and three-dimensional objects
- Teach operations on vector-valued functions, including line and surface integrals, as well as the concepts of curl and divergence, and their applications in engineering

Course Outcomes:

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

CO	Course Outcomes	BTL
CO1	Solve the systems of equations and analyze engineering problems using linear algebra techniques	K3
CO2	Apply the convergence tests of an infinite series to solve engineering problems	K3
CO3	Use differential calculus to solve optimization problems and analyze rates of change in engineering applications	K3
CO4	Calculate areas and volumes using double and triple integrals.	K3
CO5	Apply vector calculus concepts to solve problems involving work done by force fields and analyze related physical phenomena	K3

Course articulation matrix

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

(1-Low, 2 -Medium, 3-High)

Course Content	
Unit-I	[T2]
Linear Algebra Rank of a matrix (Echelon form), Finding the inverse by Gauss-Jordan method, System of linear equations: Homogeneous and Non-Homogeneous, Linear transformations, Orthogonal transformation, Eigen values and Eigenvectors, Reduction to Diagonal form	
Unit-II	[T2]
Infinite Series Infinite Sequence (Definition), Infinite Series, Comparison Tests, Integral Test, Ratio and Root Test, Alternating Series, Absolute and Conditional convergence.	
Unit-III	[T2]
Differential Calculus Mean value theorems: Rolle's theorem (without proof), Lagrange's mean value theorem (without proof), Taylor's and Maclaurin's theorems with Lagrange's form of remainder (without proof), Expansions of functions: Taylor's and Maclaurin's series Functions of Several Variables: Maxima and Minima of functions of two variables, Lagrange's method of undetermined multipliers	
Unit-IV	[T2]
Multiple Integral Double integrals (Cartesian coordinates), Change of order of integration, Triple integrals, Change of variables to polar, cylindrical and spherical coordinates, Areas as double integration and Volumes as triple integration	
Unit-V	[T2]
Vector Calculus Introduction to Gradient of a scalar field, Divergence and Curl of a vector field, Line integral, Green's theorem in the plane (without proof), Surface integrals, Stoke's theorem (without proof) and Gauss divergence theorem (without proof)	
Text Books: <ol style="list-style-type: none"> 1. Weir Maurice D., Hass Joel & Giodano Frank R. (2013). <i>Thomas' Calculus</i>. (11th Edition). Pearson Education,inc.. 2. Grewal B. S. (2017). <i>Higher Engineering Mathematics</i>. (44th Edition). Khanna Publishers. 	
Reference Books: <ol style="list-style-type: none"> 1. Kreyszig Erwin. (2013). <i>Advanced Engineering Mathematics</i>.(9th Edition).Wiley Publishers. 2. Ramana B.V.(2007). <i>Higher Engineering Mathematics</i>.Tata Mc.Graw Hill 	
Web Resources: <p>MIT OpenCourseWare: Linear Algebra https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/ NPTEL: Linear Algebra https://archive.nptel.ac.in/courses/111/104/111104137/ MIT OpenCourseWare: Multivariable Calculus https://ocw.mit.edu/courses/18-02sc-multivariable-calculus-fall-2010/ NPTEL: Engineering Mathematics-I https://archive.nptel.ac.in/courses/111/105/111105121/</p>	

24MA101 LINEAR ALGEBRA, SERIES AND CALCULUS											
Course Category:	HS	Credits:	1C								
Course Type:	Integrated Lab	Lecture-Tutorial-Practice:	0L-0T-2P								
Pre-requisites:	Mathematics-I	Continuous Assessment:	40								
		Summative Assessment:	60								
		Total Marks:	100								
Course Description: An overview of the functions of MathWorks Symbolic Math Tool Box to solve problems in Matrix algebra, differential calculus, multiple integrals, vector calculus and to test the nature of series.											
Course Objectives:											
<ol style="list-style-type: none"> 1. Find inverse, rank, eigen values, eigen vectors of a matrix and solution of system of linear equations. 2. Analyze the convergence and divergence of infinite series. 3. Find ordinary and partial derivatives 4. Expand functions as Taylor's and Maclaurin's series. 5. Determine extreme values of multi-variable function with and without constraints. 6. Calculate areas and volumes using double and triple integrals. 7. Find gradient, divergent and curl. 											
Course Outcomes: At the end of the course, the students will be able to											
CO	Course Outcomes:		BTL								
CO1	Find rank, eigen values, eigen vectors of a matrix, solution of system of linear equations.		(K3)								
CO2	Examine the convergence/divergence of infinite series		(K3)								
CO3	Determine derivatives, series expansion and extreme values of a function		(K3)								
CO4	Calculate areas and volumes using double and triple integrals		(K3)								
CO5	Find gradient of scalar field, divergence and curl of vector field		(K3)								
Course articulation matrix											
Contribution of Course Outcomes towards achievement of Program Outcomes (1 – Low, 2-- Medium, 3 -- High)											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	2			1						
CO2	2	2			1						
CO3	2	2			1						
CO4	2	2			1						
CO5	2	2			1						
Course Content											
Unit-I											
inv() function to find inverse of a matrix											
rank() function to find rank of a matrix											
rref() function to solve system of linear equations											
eig() function to find eigen values and eigen vectors of a matrix											
Unit-II											
vpa() function to evaluate numerically each term of series											
symsum() function to test the nature of the series											

Unit-III

diff() function to find ordinary and partial derivatives
taylor() function to expand functions as Taylor's and Maclaurin's series
fmincon() function to find minimum of a function with constraints

Unit-IV

int() function to find double and triple integrals

Unit-V

gradient() function to find gradient of a scalar point function
divergence() function to find divergence of a vector point function
curl() function to find curl of a vector point function

Text Books:

Matlab Lab Manual

Web Resources:

- MathWorks Linear Algebra Documentation
[Linear Algebra - MATLAB & Simulink - MathWorks India](#)
- MathWork Symbolic Calculus Tool Box
[Calculus - MATLAB & Simulink - MathWorks India](#)
- MathWork Symbolic Math Tool Box
[Symbolic Math Toolbox Documentation - MathWorks India](#)
- MathWork Optimization Tool Box
[Optimization Toolbox Documentation - MathWorks India](#)
- MathWork Symbolic Calculus Tool Box
[Calculus - MATLAB & Simulink - MathWorks India](#)

24PH101
APPLIED PHYSICS

Course Category: Physics (PH)	Basic Sciences	Credits:	3
Course Type:	Theory	Lecture-Tutorial-Practice:	3-0-0
Pre-requisites:	10+2 Physics	Continuous Assessment: Summative Assessment: Total Marks:	40 60 100

Course Description:

Applied Physics is designed to provide a comprehensive understanding of the fundamental principles of physics and their practical applications in engineering contexts. This course provides a key insight to the principles of lasers, optical fibers, semiconductors, and basic electronics, quantum mechanics, quantum computing. Through a combination of theoretical lectures and problem-solving sessions, students will develop the necessary skills to analyze and solve engineering problems.

Course Objectives:

1. Elucidate the basic concepts of Lasers and Optical fibers and their diverse applications in Science and Technology.
2. Explain the basic concepts of semiconductors with respect to energy bands and their applications.
3. Discuss the basic concepts of electronics in the design and fabrication of diode and transistors.
4. Explain the basic concepts of quantum mechanics and formalism of Schrodinger's equations.
5. Describe the principles of quantum computing and its applications in the present scenario.

Course Outcomes:

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

CO	Course Outcomes	BTL
CO1	Identify different types of Lasers and Optical fibers and their applications	K2
CO2	Apply the concepts of semiconductors towards the classification of materials based on energy bands	K2
CO3	Illustrate the formation of PN junction diode and transistor, and their applications	K2
CO4	Describe the basic concepts of quantum mechanics and its mathematical frame	K3
CO5	Summarize the basic principles of quantum computing and quantum algorithms	K3

Course articulation matrix

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

(1-Low, 2 -Medium, 3-High)

Course Content
Unit-I
<p>Lasers: Introduction, characteristics of laser, basic principles of lasers (absorption, spontaneous emission, and stimulated emission), requirements of lasers, different types of lasers: solid-state lasers (Ruby), gas lasers, (He-Ne), applications of lasers.</p> <p>Fiber Optics: Introduction, fundamentals of optical fiber, propagation of light through optical fiber, types of optical fibers, numerical aperture, fractional refractive index change, fiber optics in communication and its advantages.</p>
Unit-II
<p>Semiconductor Physics: Introduction, formation of energy bands, classification of crystalline solids, fermi level in intrinsic semiconductors, fermi level in extrinsic semiconductors, large band gap semiconductors, drift and diffusion currents, Einstein's equation, Hall effect and its applications.</p>
Unit-III
<p>Basic Electronics: Introduction, PN-junction diode, current – voltage characteristics of P N-junction diode under forward bias and reverse bias, Zener diode, bipolar junction transistor, biasing of pnp and npn, characteristics of common base, common collector and common emitter configurations, Transistor as an amplifier.</p>
Unit-IV
<p>Quantum Mechanics: Dual nature of light, matter waves and debroglie's hypothesis, G. P. Thomson experiment, Heisenberg's uncertainty principle and its application (non – existence of electron inside nucleus), Schrödinger's time independent wave equation, physical significance of wave function, particle in a one-dimensional box.</p>
Unit-V
<p>Quantum Computing: Sustainability of quantum system for information processing – classical Bits and Qu – Bits – Bloch's sphere – quantum gates – multiple Qu-Bits – advantages of quantum computing over classical computation</p>
<p>Text Books:</p> <ol style="list-style-type: none"> 1. Avadhanulu, M. N. (2019). <i>A textbook of engineering physics</i> (11th ed.). S. Chand Publishing. 2. Halliday, D. Resnick, R., & Walker, J. (2020). <i>Fundamentals of physics</i> (10th ed.). John Wiley & Sons.
<p>Reference Books:</p> <ol style="list-style-type: none"> 1. Pandey, B. K., & Chaturvedi, S. (2021). <i>Engineering Physics</i> (1st ed.). Cengage Learning. 2. Sharma, S., & Sharma, J. (2018). <i>Engineering Physics</i> (1st ed.). Pearson Education India. 3. Srinivasan, M. R. (2009). <i>Physics for Engineers</i> (1st ed.). New Age International. 4. Vijay Kumar, K. (2011). <i>Engineering Physics</i> (1st ed.). S. Chand Publishing. 5. Mc Mahon, D. (2007). <i>Quantum Computing Explained</i> (1st ed.). John Wiley & Sons. Nielsen, M. A., & Chuang, I. L. (2001). <i>Quantum computation and quantum information</i> (1st ed.). Cambridge University Press.
<p>Web Resources:</p> <ol style="list-style-type: none"> 1. Introduction to LASER, NPTEL – M. R. Shenoy, Professor, IIT – Delhi, https://onlinecourses.nptel.ac.in/noc21_ph01/preview 2. Fiber Optics, NPTEL – Vipul Rastogi, Professor, IIT – Roorkee, https://onlinecourses.nptel.ac.in/noc20_ph07/preview 3. Introduction to Semiconductor Devices, NPTEL – Naresh Kumar Emani, Professor, IIT – Hyderabad, https://archive.nptel.ac.in/courses/108/106/10810610=81/.

4. Fundamentals of Semiconductor Devices, NPTEL – Digbijoy N. Nath, IISc – Bangalore, <https://archive.nptel.ac.in/courses/108/108/108108122/#>.
5. Quantum Mechanics, NPTEL – S. Lakshmi Bala, Professor, IIT – Madras, <https://nptel.ac.in/courses/115106066>.
6. Introduction to Quantum Computing: Quantum Algorithms and Qiskit, NPTEL – Prabha Mandayam, Professor, IIT – Madras, <https://archive.nptel.ac.in/courses/106/106/106106232/>

24EN101
COMMUNICATIVE ENGLISH

Course Category:	HS	Credits:	4.5C
Course Type:	Theory	Lecture-Tutorial-Practice:	3L-0T-3P
Pre-requisites:	Intermediate English	Continuous Assessment:	40
		Summative Assessment:	60
		Total Marks:	100

Course Description:

Communicative English course covers Intercultural Communication, AI in higher education , and Natural Language Processing. It emphasizes effective listening, presentation techniques, innovation, creativity, advanced reading, and technical writing skills, addressing verbal and nonverbal communication fundamentals, processes, barriers, and strategies for confidence and fluency, preparing students for career readiness and leadership.

Course Objectives:

1. Explain the verbal and nonverbal processes of Communication.
2. Familiarize active listening skills and explain the traits of a good listener.
3. Introduce speaking skills for effective presentations.
4. Teach reading techniques to enhance employability skills.
5. Instruct writing techniques to enhance employability skills.

Course Outcomes:

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

CO	Course Outcomes:	BTL
CO1	Relate and recall the verbal and nonverbal processes of Communication.	(K1)
CO2	Demonstrate active listening skills and acquire the traits of good listener	(K2)
CO3	Apply English speaking skills to deliver effective presentations	(K3)
CO4	Appraise reading techniques like skimming and scanning to enhance employability skills	(K4)
CO5	Assess writing techniques like outlining and clustering to enhance employability skills	(K5)

Course articulation matrix

Contribution of Course Outcomes towards achievement of Program Outcomes (1– Low, 2-- Medium, 3 --High)

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

Course Content

Unit-I

Introduction to Communication:

Basics of Communication, Process of Communication, Levels of Communication, Verbal and Nonverbal Communication, Channels of Communication, Barriers to Communication, *Intercultural Communication*.

Unit-II

Listening Skills:

Effective Listening, Active Traits of a Good Listener, Listening Modes, Types of Listening, Barriers to Effective Listening, Listening for general content, *AI in Higher Education: Opportunities, Challenges, and Ethical Considerations*.

Unit-III

Speaking Skills:

Achieving Confidence, Clarity & Fluency, Effective Presentation Strategies, Meetings, Conferences, Paralinguistic Features, Barriers to Speaking, *Natural Language Processing*.

Unit-IV

Reading Skills:

Reading Techniques, Techniques for Good Comprehension, Study Skills, Reading and Interpretation, Intensive, Extensive & Critical Reading, Reading for Different Purposes, *Innovation and Human Creativity*.

Unit-V

Writing Skills:

Technical Writing, Importance and Characteristics, Techniques for Good Technical Writing ,Paragraph Construction, Essays, Reports, *Communication for Career-readiness and Leadership*.

Text Books:

1) Raman,M., & Sharma,S.(2012).*Technical Communication* (2nded.). Oxford University Press.

Reference Books:

- 1) Pushplata, & Kumar, S.(2011).*Communication Skills*. Oxford University Press.
- 2) Rizvi,A.(2005).*Effective Technical Communication*. Tata Mc Graw- Hill Publishing Company Limited.
- 3) Mishra,S, & Muralikrishna,C.(2006). *Communication Skills for Engineers*. Pearson.

Web Resources:

http://ndl.iitkgp.ac.in/he_document/ekumbh/ekumbh/85?e=5|text%20books%20on%20technical%20communication|| Kulbhushan Kumar. *English for Technical Professionals*.

<https://www.gtuelibrary.edu.in/publication/Technical%20communication%205th%20June'09.pdf>Prof. (Dr.) M. D. Desai, *Technical Communication*.

<https://gnindia.dronacharya.info/CSE-AI-ML/Common-Subjects/Downloads/Technical-Communication/Books/Technical-Communication-Book-1.pdf>

Technical Communication Principles and Practice

Meenakshi Raman and Sangeeta Sharma

Laboratory Exercises

1) Phonetics.

Introduction – Articulatory Phonetics (The Sounds of Speech Production) – Phonological Concepts (Sound Patterns and Rules) – Practical Applications.

2) Listening Comprehension

Introduction – Strategies for Effective Listening – Types of Listening (Discriminative, Comprehensive, Evaluative, and Appreciative) – Improving Listening Comprehension (Note-taking and Summarizing) – Overcoming Distractions and Misunderstandings – Practice Activities.

3) Conversation Starters- Introducing Self and others

Icebreakers – Introducing Yourself – Introducing Others – Accepting – Declining – Agreeing – Disagreeing – Seeking Clarification – Seeking Permission – Expressing Interest – Offering Help.

<p>4) Just A Minute Introduction – Rules of the Game (No Hesitation, No Deviation, No Repetition)– Strategies for Success – Benefits of JAM – Common Challenges – Practice Sessions.</p>
<p>5) Situational Dialogues & Role Plays Introduction – Benefits – Planning and Structuring a Role Play – Everyday Scenarios – Tips for Effective Role Playing – Role Play Practice.</p>
<p>6) Reading Comprehension Understanding Reading Comprehension – Strategies for Effective Reading – Types of Reading Comprehension Questions – Improving Vocabulary and Context Clues – Practice Exercises – Techniques to Master.</p>
<p>7) Vocabulary Vocabulary Building –Exploring Prefixes– Discovering Suffixes– Uncovering Roots– Selected Word List (500 Essential Words for Vocabulary Development) – Practical Application.</p>
<p>8) Grammar Introduction to Grammar Essentials – Understanding Tenses – Mastering Articles – Navigating Prepositions – Practice Exercises – Tips for Improving Grammar Skills.</p>
<p>9) Describing People, Places, Objects and Events Introduction to Descriptive Language –Describing People – Describing Places – Describing Objects – Describing Events – Practice Activities.</p>
<p>10) Poster Presentation Introduction – Designing an Effective Poster – Content Development – Presenting Information Visually – Effective Delivery – Tips for Engaging Your Audience.</p>
<p>11) Letter Writing Introduction – Structure of a Letter – Types of Letters – Writing Techniques – Addressing Envelopes –Practice Activities.</p>
<p>12) Email Writing Introduction – Parts of an Email – Tone, Style and Language – Email Etiquette – Dos and Don'ts of Professional Email Communication – Organizing Inbox and Responding Promptly – Practice Activities.</p>
<p>Textbook: Communicative English Lab Manual</p>
<p>References: 1) D. Sudha Rani. (2011). <i>A Manual for English Language Laboratories</i>. Pearson, Noida, 2) Nira Konar. (2011). <i>English Language Laboratories</i>. PHI Learning Private Limited, New Delhi.</p>
<p>English Language Communication Skills Lab Software: 1) Walden 2) Softx 3) Visionet Spears Digital Language Lab software Advance Pro 4) ODll Language Learner’s Software, Orell Techno Systems</p>

COURSE CODE: 24CS102
PROGRAMMING USING C

Course Category:	Engineering Science (ES)	Credits:	3
Course Type:	Theory	Lecture -Tutorial-Practice:	3-0-0
Pre-requisites:		Continuous Evaluation:	40
		Semester end Evaluation:	60
		Total Marks:	100

Course Description

This course introduces foundational programming concepts, covering algorithms, flowcharts, and pseudo code. It explores the C language structure, data types, operators, control structures, loops, arrays, strings, and functions. Advanced topics include pointers, dynamic memory allocation, structures, unions, enumerations, and file handling. Students gain hands-on experience with inter-function communication, recursion, sorting/searching techniques, and memory management. The course emphasizes practical coding skills, problem-solving, and program design, providing a solid foundation for software development using the C language.

Course Objectives

- To introduce students to the fundamentals of computer programming.
- To provide hands-on experience with coding and debugging.
- To foster logical thinking and problem-solving skills using programming.
- To familiarize students with programming concepts such as data types, control structures, functions, and arrays.
- To encourage collaborative learning and teamwork in coding projects.
- To evaluate and apply C programming techniques proficiently for searching and sorting.

Course Outcomes

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

	Course Outcomes	BTL	POI
CO1	Understand fundamental programming concepts in C through algorithms, flowcharts, and selection statements.	K2	1.7.1,2.5.1, 2.5.2, 2.7.1
CO2	Develop efficient C programs using loops, arrays, and strings using control structures.	K3	1.7.1,2.5.1, 2.5.2, 2.6.3,3.5.1
CO3	Implement modular C programs using functions, pointers, and memory optimizations.	K3	1.7.1, 2.5.2, 2.6.3, 3.5.1, 5.4.1
CO4	Develop C programs using structures and unions for user defined data types.	K3	2.5.2, 2.6.3,3.5.1
CO5	Analyze the use of enumerations and file handling techniques in C to manage data efficiently and solve real-world programming problems.	K4	1.7.1,2.5.1, 2.5.2, 2.6.3,3.5.1

Contribution of Course Outcomes towards achievement of Program Outcomes

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

(1- Low, 2 - Medium, 3 – High)

Unit-I : Introduction to C & Problem solving

Introduction to the C Language: Introduction to Programming Languages, Basics of a Computer Program, Algorithms, flowcharts (Using Dia Tool), pseudo code. Introduction to Compilation and Execution, Background of C program, Identifiers, Types, Variables, Constants, Memory Layout, Input/Output, Programming Examples.

Structure of a C Program: Logical Data and Operators, Expressions, Precedence and Associativity, Evaluating Expressions, Type Conversion, Statements, Storage Class.

Selection: Two-way Selection, Multiway selection, More Standard Functions.

Unit-II: Repetition, Arrays & Strings

Repetition: Concept of Loops in C, Loop Examples, the Calculator Program.

Arrays: Array Concepts in C, Inter-Function Communication, Array Applications, One Dimensional Arrays, Linear Search and Binary Search Techniques, Selection Sort, Bubble Sort, Two Dimensional Arrays, Multidimensional Arrays.

Strings: String Concepts, C Strings, String Input/output Functions, Arrays of Strings, String Manipulation Functions, String- Data Conversion.

Unit-III: Functions, Pointers & Memory Allocations

Functions: Functions in C, User Defined Functions, Call by Value, Call Value Reference, Inter-Function Communication, Standard Functions, Scope, Recursion and advantages.

Pointers: Introduction to Pointers, Pointers for Inter-Function Communications, Pointers to Pointers, Compatibility, Lvalue and Rvalue. Arrays and Pointers, Pointers Arithmetic and Arrays, Passing an Array to a Function, Array of Pointers.

Memory Allocation: Need of dynamic memory allocation, malloc(), calloc(), free(), realloc(), NULL, Stack vs. Heap Allocation.

Unit-IV: Structures & Unions

Structures: Structure Type Declaration, Initialization, Accessing Structures, Operations on Structures, Complex Structures, Structures and Functions, Sending the Whole Structure, Passing individual structure members, passing structure via pointer, nested structure.

Unions: Referencing Unions, Initializers, Unions and Structures, Internet Address, Programming Applications.

Unit-V: Enumerations & Files

Enumerations: The Type Definition (Typedef), Enumerated Types: Declaring an Enumerated Type, Operations on Enumerated Types, Enumeration Type Conversion, Initializing Enumerated Constants, Anonymous Enumeration: Constants, Input/Output Operators.

Files: introduction to the files, Uses of Files, Text files Vs. Binary files, Opening and closing FILE , Modes of FILE operation, Command line arguments, Standard Library Input /Output functions ,Character i/o functions, File Handling functions.

Text Books:

1. "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988
2. Behrouz A. Forouzan and Richard F. Gilberg, "Computer Science A Structured Programming Approach Using C", CENGAGE Learning, Third Edition.
3. Schaum's Outline of Programming with C, Byron S Gottfried, McGraw-Hill Education, 1996

Reference Books:

1. Computing fundamentals and C Programming, Balagurusamy, E., McGraw-Hill Education, 2008.
2. Programming in C, Rema Theraja, Oxford, 2016, 2nd edition
3. C Programming, A Problem Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE, 3rd edition
4. Horowitz Sahni and Anderson-Freed, "Fundamentals of Data Structures in C", 2nd edition, Universities Press, 2011

Web Resources:

1. NPTEL, "NOC: Introduction to Programming in C," [Online]. Available: <https://nptel.ac.in/courses/106/104/106104128/>. [Accessed: Feb. 25, 2025].
2. Coursera, "C for Everyone: Structured Programming," [Online]. Available: <https://www.coursera.org/learn/c-structured-programming/>. [Accessed: Feb. 25, 2025].
3. GeeksforGeeks, "Arrays, Pointers and Functions in C," [Online]. Available: <https://www.geeksforgeeks.org/arrays-and-pointers-functions-in-c/>. [Accessed: Feb. 25, 2025].
4. GeeksforGeeks, "C Enumerations," [Online]. Available: <https://www.geeksforgeeks.org/enumeration-enum-c/>. [Accessed: Feb. 25, 2025].

5. Codeforwin, "Understanding malloc, calloc, realloc, and free," [Online]. Available: <https://codeforwin.org/2018/07/malloc-calloc-realloc-free-functions-c-programming.html>. [Accessed: Feb. 25, 2025].
6. GeeksforGeeks, "Unions in C," [Online]. Available: <https://www.geeksforgeeks.org/union-c/>. [Accessed: Feb. 25, 2025].
7. Programiz, "Structures in C," [Online]. Available: <https://www.programiz.com/c-programming/c-structures>. [Accessed: Feb. 25, 2025].
8. TutorialsPoint, "Formatting Input/Output Functions and Character Input/Output Functions," [Online]. Available: https://www.tutorialspoint.com/cprogramming/c_file_io.htm. [Accessed: Feb. 25, 2025].
9. Programiz, "Command-Line Arguments," [Online]. Available: <https://www.programiz.com/c-programming/c-command-line-arguments>. [Accessed: Feb. 25, 2025].
10. W3Schools, "C Programming Language," [Online]. Available: <https://www.w3schools.com/c/index.php>. [Accessed: Feb. 25, 2025].

SIDDHARTHA ACADEMY OF HIGHER EDUCATION (SAHE)

COURSE CODE: 24CS181
PROGRAMMING USING 'C' LAB

Course Category:	Engineering Science (ES)	Credits:	1.5
Course Type:	Practical	Lecture -Tutorial-Practice:	0-0-3
Pre-requisites:		Continuous Evaluation:	60
		Semester end Evaluation:	40
		Total Marks:	100

Course Description

This hands-on course introduces students to the foundational principles of the C programming language, integrating essential Linux command-line skills and compiler tools such as GCC and Turbo C. The course emphasizes problem-solving, logical thinking, and algorithm development. Students will work through a structured progression from simple input/output to complex file operations and data structures, including practical projects and assignments.

Course Objectives

The primary objective of this C Programming Lab course is to provide hands-on practical experience in writing, debugging, and executing C programs. It aims to develop students' understanding of fundamental programming concepts such as input/output operations, control structures, arrays, strings, functions, and pointers through practical exercises. By the end of the course, students will be proficient in using essential C programming tools, developing logical problem-solving skills, and applying programming concepts to build efficient programs in a Linux environment using compilers like Turbo C and GCC.

Course Outcomes

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

	Course Outcomes	BTL	POI
CO1	Understand basic Linux shell commands to compile and run C programs using GCC.	K2	1.7.1,2.5.1, 2.5.2, 2.7.1
CO2	Select the right control structure for solving the problem.	K3	1.7.1, 2.5.1,2.5.2, 2.6.3,3.5.1
CO3	Develop C programs which utilize memory efficiently using programming constructs like pointers.	K3	1.7.1, 2.5.2, 2.6.3,3.5.1 5.4.1
CO4	Develop, Debug and Execute programs to demonstrate the applications of arrays, functions, basic concepts of pointers in C.	K3	2.5.2, 2.6.3,3.5.1
CO5	Analyze the use of enumerations and file handling techniques in C to manage data efficiently and solve real-world programming problems.	K4	1.7.1, 2.5.1,2.5.2, 2.6.3,3.5.1

Contribution of Course Outcomes towards achievement of Program Outcomes

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

(1- Low, 2 - Medium, 3 – High)

Course Content:

WEEK- 1

1. Basic Linux environment and its editors like Vi, Vim & Emacs etc.
2. Exposure to Turbo C, gcc
3. Write a C program to print the following output Input:

```

Enter a Character: *
Enter a Number: 1
Output:
    
```

```

* * * * *
*      1      *
*      1      *
*      1      *
*      1      *
*     1 1 1    *
* * * * *
    
```

4. Write a C program that takes two numbers and an arithmetic operator (+, -, *, or /) as input from the user using `scanf()`, performs the corresponding operation, and displays the result using `printf()`.
5. Write a C program that reads two numbers from the user using `scanf()`, and then swaps their values using both methods:
 1. With a third variable
 2. Without using a third variable
 Finally, display the swapped values using `printf()`.

6. Declare three variables of integer data type. User has to input three valid numbers. Display Sum and average of user entered numbers. Approach is to take three numbers and find their sum and average using the formula given below-

Sum: $a+b+c$

Average: $(a+b+c)/3$

Where a,b,c are the three numbers.

Sample Input:

Enter 3 Values: 10 20 30

Sample Output: Sum: 60 Average: 20

WEEK-2

1. Input the temperature in Fahrenheit and output the equivalent temperature in Celsius and Vice – Versa. Input two numbers. The first is for a Celsius value, and the second is for a Fahrenheit value. Input Celsius value, convert it to Fahrenheit. Use the formula $F = C*9/5 + 32$ for conversion. In case of decimals, show up to 1 decimal value.

Input the Fahrenheit value, convert it to Celsius.

Use the formula $C = (F-32)*5/9$ for conversion. In case of decimals, show up to 1 decimal value.

Sample Input: 0 100

Sample Output: 32 37.7

2. You can calculate a Simple Interest by just providing the Principle Amount, Rate of Interest and Time or Periods provided by the user input. We can calculate the Simple Interest by the using the below Formula.

Simple Interest = $(\text{Principal} * \text{Rate} * \text{Time}) / 100$

Principal (P): The principal is the amount that was initially borrowed (loan) from the bank or invested.

Rate (R): It is the rate of interest at which the principal amount is given to someone for a certain time; the rate of interest can be 5%, 10%, or 13%, etc.

Time (T): Time is the duration for which the principal amount is given to someone.

Constraints: $1 \leq \text{Principal} \leq 10000$ $1 \leq \text{Rate} \leq 10$

$1 \leq T \leq 30$

Sample Input: 100 3 10

Sample Output: 30

3. You are working on a program that tracks the daily sales of a small bookstore. The owner wants to calculate the total sales and average sales for three consecutive days to better understand business trends. Write a program that takes the sales figures for three days as input, calculates the total sales, and finds the average sales. How will you implement this in C?

4. Write a C program to evaluate and display the result of the following expressions. Use the given variable declarations and initialize them with appropriate values. Also, print the results of each expression.

- $A+B*C+(D*E) + F*G$
- $A/B*C-B+A*D/3$
- $A+++B---A$
- $J= (i++) + (++i)$

5. Write a C program to calculate electricity bill according to the given condition:

For first 50 units Rs. 0.50/unit

For next 100 units Rs. 0.75/unit

For next 200 units Rs. 1.20/unit

For unit above 250 Rs. 1.50/unit

An additional surcharge of 20% is added to the bill.

WEEK- 3

1. To find factorial of the any given positive number. The factorial of a positive number n is given by: $1*2*3*4 \dots$ Note: This program should take a positive integer from the user as the factorial of a negative number doesn't exist and, the factorial of 0 is 1. Compute the factorial using any loop. Since the factorial of a number may be very large, the type of factorial variable is declared as unsigned long. If the user enters a negative number, the program should display a custom error message.

2. Input a number, check the given number is a prime or not. A prime number should be a natural number greater than 1 that has no positive divisors other than 1 and itself.

Test Data and Output: Enter n: 5

Output: Prime

Enter n: 6

Output: Not Prime

3. Checking a number palindrome. Number should be a positive integer having more than one digit as all the single digits are palindromes.

Test Data and output:

Input : 2002 Input: 1234

Output: true Output: false

4. Construct a pyramid of numbers. A pyramid of numbers represents the number of individuals per unit area of various trophic areas where producers are kept at the base and the tip is occupied by top carnivores.

The pyramid of numbers is mostly upright. The members of successive higher trophic levels are higher than the previous one.

1. A higher trophic level has fewer individuals than that of the lower trophic levels.

1

2 2 2

3 3 3 3 3

4 4 4 4 4 4 4

5 5 5 5 5 5 5 5

5. Develop a C program that takes an integer as input from the user and prints 'Yes' if the number is an Armstrong number, and 'No' otherwise. (This adds specific input/output requirements.)

WEEK- 4

1. Write a C program to delete all duplicate elements from an array.

The program should:

- Ask the user to enter the number of elements in the array.
- Accept the array elements from the user.
- Remove all duplicate values from the array.
- Display the new array with only unique elements.

2. Write a C program to insert an element into a specific position in an array.

The program should:

- Ask the user to enter the size of the array and its elements.
- Ask the user to enter the element to insert and the position at which it should be inserted.
- Insert the element at the specified position by shifting the existing elements.
- Display the updated array.

3. Find 2's complement of the given binary number.

To get 2's complement of a binary number, simply invert the given number and add 1 to the least significant bit (LSB) of given result.

Test Data and Output:

Find 2's complement of binary number 10101110.

Simply invert each bit of given binary number, which will be 01010001. Then add 1 to the LSB of this result, i.e., $01010001+1=01010010$ which is answer.

Find 2's complement of binary number 10001.001.

Simply invert each bit of given binary number, which will be 01110.110 Then add 1 to the LSB of this result, i.e., $01110.110+1=01110.111$ which is answer.

4. Write a C program to sort array elements in ascending or descending order.

5. Given an array of integers **nums** and an integer **target**, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution and you may not use the same element twice. You can return the answer in any order.

Sample Input:

Enter the size of an array: 4

Enter array elements: 2 7 11 15

Enter target: 9

Output: [0,1]

WEEK- 5

1. A matrix can only be added to another matrix if the two matrices have the same dimensions. To add two matrices, just add the corresponding entries, and place this sum in the corresponding position in the matrix which results.

Input elements in 3x3 matrix1: 1 2 3

4 5 6

7 8 9

Input elements in 3x3 matrix2: 9 8 7

6 5 4

3 2 1

Sum of both matrix = 10 10 10

10 10 10

10 10 10

2. Matrix multiplication is a binary operation that produces a matrix from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the **matrix product**, has the number of rows of the first and the number of columns of the second matrix. The product of matrices **A** and **B** is denoted as **AB**.

3. Concatenate two strings without using built-in functions

Note: User would be asked to enter two strings and then the program would concatenate them. For concatenation we have not used the standard library function `strcat()`, instead we have written a logic to append the second string at the end of first string.

Test Data and Output:

Str1: Good Str2: Morning Output: Good Morning

4. Reverse a string using built-in and without built-in string functions

Using built-in function: The function is used for reversing a string. The reversed string will be stored in the same string.

5. Input two strings **str1** in lowercase, **str2** in uppercase. Print the lower case string **str1** in uppercase and the uppercase string **str2** in lowercase.

WEEK- 6

1. Write a C program using a user-defined function to find the biggest number given any three numbers.
2. Write a C **program** to calculate the **factorial of a given number** using recursion. The program should prompt the user to enter a number and then compute its factorial using a recursive function.
3. Write a C **program** to generate the **Fibonacci series up to N terms** using a recursive function. The program should prompt the user to enter a number N, then compute and display the first N **Fibonacci numbers**.
4. Write a C **program** to swap two numbers using **call by value and call by reference**. The program should demonstrate that changes made inside the function do not affect the original values in the main function.
5. How can recursion be used to calculate NCR? Write a C program to demonstrate this.

WEEK- 7

1. String is a sequence of characters. Input two strings str1 and str2. Copy the contents of str1 to str2 using functions and pointers. Define a function copystr() with two pointer arguments copystr(*str1,*str2) Approach :

1. Scan string str from 0 to length-1.
2. check one character at a time based on ASCII values
 - if(str[i] >= 65 and str[i] <=90), then it is uppercase letter,
 - if(str[i] >= 97 and str[i] <=122), then it is lowercase letter,
 - if(str[i] >= 48 and str[i] <=57), then it is number,
 - else it is a special character

3. Print all the counters

Sample Input:

Enter a string: Programming

Sample Output:

String1: Programming

String2: Programming

2. Given a string, write a program to count the occurrence of Lowercase characters, Uppercase characters, Special characters, and Numeric values. Define a function to count.

Input : #CseAi01dOr@gAIml07

Output :

Upper case letters : 5

Lower case letters : 8

Numbers : 4

Special Characters : 2

Input : *AiMlC4AiDsS*

Output :

Upper case letters : 6

Lower case letters : 4

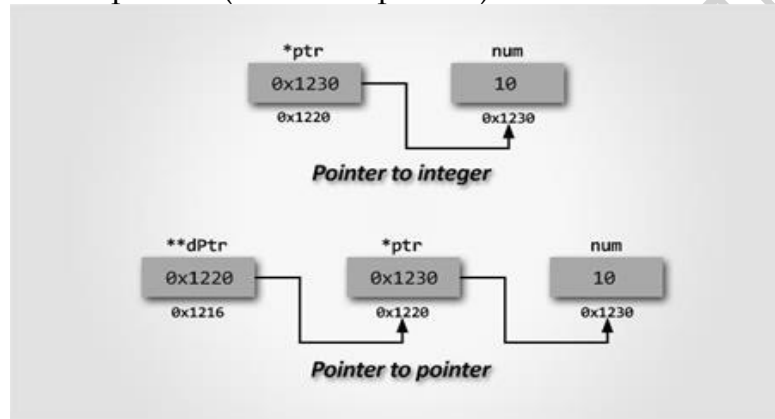
Numbers : 1

Special Characters : 2

3. Write a c program to perform simple pointer arithmetic operations in c.

4. Write a c program to find sum of array elements by using array of pointers in c

5. When a pointer holds the address of another pointer then such type of pointer is known as pointer-to-pointer or double pointer. ptr is a normal pointer that holds the address of an integer variable num. There is another pointer **dptr in the diagram that holds the address of another pointer ptr, the pointer dptr here is a pointer-to-pointer (or double pointer). Demonstrate double pointers using the



following representation.

WEEK- 8

1. Write a C program to define a structure named Student that contains the following members:

- name (character array)
- rollNumber (integer)
- age (integer)

Using a structures concept, store and display the details of two students.

2. Write a C program to define a structure named Student with the following members:

- name (character array) - to store the student's name.
- rollNumber (integer) - to store the roll number.
- marks (array of 3 integers) - to store marks obtained in 3 subjects.

Using an array of structures, read the details of two students from the keyboard. Calculate and display:

- Name of the student
- Roll number
- Marks in 3 subjects
- Total marks obtained

3. Write a C program to define a structure named **Book** with the following members:

- title (character array) – to store the book's title.
- author (character array) – to store the author's name.
- price (float) – to store the price of the book.

Using an array of structures, read the details of **five books** from the keyboard and display them in a formatted output.

4. Write a C program to define a structure **Student** that includes the following details:

- Name (character array)
- Roll Number (integer)
- Age (integer)
- Date of Birth (another structure with day, month, and year as integers)

Note: Student Details (with Date of Birth as a Nested Structure)

5. Write a C program to define a structure **Employee**, which contains the following details:

- **Employee Name** (character array)
- **Employee ID** (integer)
- **Salary** (float)

Additionally, the structure **Employee** should have a **nested structure Address**, which includes:

- **City** (character array)
- **State** (character array)
- **Pin Code** (integer)

The program should:

- Accept details of an employee, including their address.
- Display the entered details in a proper format.

Write the complete C program, including structure definition, input, and output statements.

Note: Employee Records (with Address as a Nested Structure)

WEEK- 9

1. Write a C program to demonstrate **passing individual structure members** to a function. Define a structure for a student with members: name, roll number, and marks. Pass individual members to a function to display the details.

2. Explain the concept of **passing an entire structure** to a function in C. Write a program to define a structure for an Employee with members: name, ID, and salary. Pass the entire structure to a function to display the details.
3. Define a union Data with members: i (integer), f (float), str (string). Write a program to assign and print values to each member of the union one by one. Observe what happens to the values.
4. Explain the concept of **passing a structure through pointers** in C. Write a program to define a structure for a Student with members: name, roll number, and marks. Pass the address of the structure to a function and display the student details using pointers.
5. Write a C program to **add two complex numbers**.
 - Define a **structure** Complex with two members: real (to store the real part)
 - imag (to store the imaginary part)
 - Take input for two complex numbers from the user.Perform the addition of the two complex numbers: Add the **real parts** separately.
 - Add the **imaginary parts** separately.
 - Display the **sum** of the two complex numbers in the form **(a + bi)**.

Sample output:

Enter first complex number: 3 2

Enter second complex number: 1 7

Sum: 4 + 9i

WEEK- 10

1. Write a C program that defines an enumeration for different car brands and asks the user to enter a number. Display the corresponding car brand.
2. Create an enum for traffic signals (Red, Yellow, Green). Write a function that takes an enum value as input and prints its meaning (e.g., Red → Stop).
3. Write a C program to store and process student data using dynamic memory allocation with calloc(). The program should:
 - Allow the user to enter the number of students (n).
 - Dynamically allocate memory for n students using calloc().
 - For each student, input details such as **roll number, name, and marks**.
 - Display the list of students who have **failed** (consider marks less than 40 as failed).
4. Write a C program to print the corresponding weekday name for a given integer value (1 to 7) using enumeration constants. The program should define an enum for the weekdays, take an integer input from the user, and display the corresponding weekday name. If the input is out of range, display an appropriate error message.

5. Write a C program to write and read text into a file. The program should prompt the user to enter a string and write it to a text file. Then, it should open the same file, read the content, and display it on the screen.

6. **Write a C program to copy the contents of one file to another using command-line arguments.** The program should accept the source filename and the destination filename as command-line arguments. It should read the contents from the source file and write them to the destination file. If the source file does not exist or an error occurs, display an appropriate error message.

Web Resources:

1. LinuxCommand.org, "Basic Linux Environment, Editors, Turbo C, GCC," [Online]. Available: <https://linuxcommand.org/>. [Accessed: Feb. 25, 2025].
2. Programiz, "Pattern Printing & Block Letters in C," [Online]. Available: <https://www.programiz.com/c-programming/examples/pattern-printing>. [Accessed: Feb. 25, 2025].
3. GeeksforGeeks, "Arithmetic Operations & Swapping Numbers," [Online]. Available: <https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/>. [Accessed: Feb. 25, 2025].
4. CodingCompiler, "Sum and Average of Numbers," [Online]. Available: <https://codingcompiler.com/c-program-find-sum-average-three-numbers/>. [Accessed: Feb. 25, 2025].
5. TutorialsPoint, "Reading and Printing Multiple Integers," [Online]. Available: https://www.tutorialspoint.com/cprogramming/c_arrays.htm. [Accessed: Feb. 25, 2025].
6. Programiz, "Mixed Data Types Input & Format Specifiers," [Online]. Available: <https://www.programiz.com/c-programming/c-input-output>. [Accessed: Feb. 25, 2025].
7. GeeksforGeeks, "Files in C," [Online]. Available: <https://www.geeksforgeeks.org/basics-file-handling-c/>. [Accessed: Feb. 25, 2025].

24UC181
DESIGN THINKING
(CE, ECE, EEE, EIE & ME)

Course Category:	Engineering Sciences (ES)	Credits:	1
Course Type:	Laboratory	Lecture-Tutorial-Practice:	0-0-2
Pre-requisites:		Continuous Assessment:	60
		Summative Assessment:	40
		Total Marks:	100

Course Description:

This course introduces first-year undergraduate engineering students to the principles and practices of design thinking, emphasizing user-centred problem-solving and innovation. Students will learn to empathize with users, define problems, ideate solutions, develop prototypes, and test these solutions. The course will also explore the application of design thinking across various engineering disciplines, including Civil Engineering (CE), Electronics and Communication Engineering (EC), Electrical Engineering (EE), Mechanical Engineering (ME), and Information Technology (IT).

Course Objectives:

1. To introduce the fundamental concepts and process of design thinking.
2. To develop students' ability to empathize with users and accurately define engineering problems.
3. To encourage creativity and the generation of a wide range of ideas.
4. To enable students to prototype and test their design solutions.
5. To illustrate the application of design thinking to solve core engineering design problems.

Course Outcomes:

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

CO	Course Outcomes	BTL
CO1	Explain the principles of design thinking.	K2
CO2	Apply empathy and problem-defining techniques in engineering contexts.	K3
CO3	Employ innovative ideas using various ideation techniques.	K3
CO4	Model and test prototypes to validate engineering solutions.	K4
CO5	Practice design thinking strategies in civil, electronic, electrical, mechanical, and IT engineering projects	K3

Course articulation matrix

COs	POs										
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1								3	3		3
CO2								3	3		3

CO3			3		3	3		3	3		3
CO4					3			3	3		3
CO5					3			3	3		3

(1-Low, 2 -Medium, 3-High) (WK8)

Course Content

Unit-I

Description: This unit introduces the foundational concepts of design thinking and its importance in solving complex engineering problems.

Introduction to Design Thinking: Introduction to Design Thinking, History and Evolution of Design Thinking, Key Principles and Mindsets, The Design Thinking Process: An Overview, The Role of Design Thinking in Engineering.

Examples/Applications/Case Studies:

- Case Study: How IDEO uses design thinking to innovate.
- Example: Apple's design thinking approach in product development.

Lab/Practice Exercises:

1. Identifying and analyzing engineering problems.
2. Group discussions on real-world design thinking examples.
3. Role-playing to understand user perspectives.

Unit-II

Description: This unit focuses on understanding users through empathy and accurately defining engineering problems to guide the design process.

Empathize and Define: The Importance of Empathy in Design Thinking, Techniques for Empathy: Interviews, Observations, and Surveys, Defining the Problem: Point of View Statements, Tools for Problem Definition: Journey Maps, Personas, Translating Empathy into Insights.

Examples/Applications/Case Studies:

- Example: Empathy maps in understanding customer needs.
- Case Study: How Airbnb uses empathy in design thinking.

Lab/Practice Exercises:

1. Conducting user interviews.
2. Creating personas and journey maps.
3. Developing problem statements.

Unit-III

Description: This unit explores creative techniques for generating a wide range of ideas to solve defined engineering problems.

Ideate: The Ideation Phase in Design Thinking, Brainstorming Techniques, Creative Thinking and Innovation, Selecting and Refining Ideas, Overcoming Creative Blocks.

Examples/Applications/Case Studies:

- Example: Brainstorming session in a tech startup.
- Case Study: Google's "20% time" policy for innovation.

Lab/Practice Exercises:

1. Group brainstorming sessions.
2. Idea selection and refinement exercises.
3. Sketching and visualizing ideas.

Unit-IV

Description: This unit covers the creation of prototypes and the methods for testing and

iterating on these prototypes to develop viable engineering solutions.

Prototype and Test: Prototyping Basics and Importance, Types of Prototypes: Low-fidelity vs. High-fidelity, Tools and Materials for Prototyping, Testing Prototypes: Methods and Techniques, Iterating Based on Feedback.

Examples/Applications/Case Studies:

- Example: Prototyping in the automotive industry.
- Case Study: Iterative prototyping at Dyson.

Lab/Practice Exercises:

1. Building low-fidelity prototypes.
2. Conducting user testing sessions.
3. Analyzing feedback and iterating designs

Unit-V

Description: This unit examines the application of design thinking across various engineering disciplines, including CE, EC, EE, ME, and IT, to solve engineering design problems.

Design Thinking in Civil Engineering (CE) , Applying Design Thinking in Electronics and Communication Engineering (EC), Design Thinking for Electrical Engineering (EE) , Innovation in Mechanical Engineering (ME), Design Thinking in Information Technology (IT).

Examples/Applications/Case Studies:

- Case Study: Design thinking in sustainable building design (CE).
- Example: Innovation in wearable technology (EC).
- Case Study: Smart grid solutions (EE).
- Example: Prototyping in automotive design (ME).
- Case Study: User-centric software development (IT).

Lab/Practice Exercises:

1. Developing design thinking strategies for a civil engineering project.
2. Prototyping an electronic device solution.
3. Creating user-centric IT solutions

Text Books:

1. Brown, T. (2009). Change by Design: How Design Thinking Creates New Alternatives for Business and Society. Harper Business.
2. Cross, N. (2011). Design Thinking: Understanding How Designers Think and Work.
 1. Berg.

Reference Books:

1. Liedtka, J., & Ogilvie, T. (2011). Designing for Growth: A Design Thinking Tool Kit for Managers. Columbia University Press.
2. Kelley, T., & Kelley, D. (2013). Creative Confidence: Unleashing the Creative Potential Within Us All. Crown Business.
3. Martin, R. L. (2009). The Design of Business: Why Design Thinking is the Next Competitive Advantage. Harvard Business Press.

Web Resources:

1. https://onlinecourses.swayam2.ac.in/aic26_ge05/course

**24PH181
PHYSICS LAB**

Course Category:	Programme Core (PC)	Credits:	1
Course Type:	Practical	Lecture-Tutorial-Practice:	0-0-2
Pre-requisites:	10+2 Physics	Continuous Assessment:	60
		Summative Assessment:	40
		Total Marks:	100

Course Description:

The Engineering Physics laboratory for B. Tech students is designed to provide hands on experience in basic concepts of optics, lasers, optical fibers, waves and oscillators, electricity and magnetism, quantum mechanics, and semiconductors. This course aims to foster the practical skills essential through experimentation, measurement, and analysis for the students of computer science allied branches like CSE, AI-ML, AI-DS, and Information Technology, ECE, EEE, EIE, CE and ME branches. Students will work with advanced equipment under the guidance of experienced faculty to explore various physical phenomena and their engineering applications.

Course Objectives:

1. Demonstrate the basic concepts of wave optics and the experimental evidence of wave nature of light by interference and diffraction phenomena.
2. Provide the experimental knowledge of understanding the properties of semiconductors and their applications.
3. Elucidate the basic concepts of Lasers and Optical fibers and their diverse applications in Science and Technology.

Course Outcomes:

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

CO	Course Outcomes	BTL
CO1	Identify the wave nature of light by the concepts like interference and diffraction of light.	K4
CO2	Distinguish the semiconductors based on carrier concentration and Hall coefficient.	K4
CO3	Calculate the specific charge of an electron and work function of a photocell.	K4
CO4	Demonstrate a comprehensive understanding of wave phenomena, resonance, and their application across various physical systems.	K3
CO5	Apply the laser principles, optical fiber transmission characteristics, and their practical applications in telecommunications and photonics.	K4

Course articulation matrix

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

(1-Low, 2 -Medium, 3-High)

Course Content with CO mapping

A. Experiments common to all branches:

1. Solar cell – Determination of fill factor (Common to all branches). (CO2)
2. Hall effect – Hall coefficient measurement (Common to all branches). (CO2)

B. Experiments for CSE, IT, AIDS, and AIML branches:

3. Newton’s Rings – Radius of Curvature of a plano convex lens (CO1)
4. Diffraction grating – Wavelength of laser light (CO1,CO5)
5. Photocell – Study of V – I characteristics and determination of work function (CO3)
6. Compound pendulum – Measurement of ‘g’ (CO4)
7. Specific charge (e/m) of an electron – J. J. Thomson method (CO3)
8. AC Sonometer – Verification of vibrating laws (CO4)
9. Optical Fiber – Determination of Numerical Aperture (CO5)
10. Figure of Merit of a Galvanometer (CO4)
11. Variation of Magnetic field along the axis of current – carrying circular coil (CO3)
12. Diffraction grating – Measurement of wavelength of mercury source (CO1,CO5)

C. Experiments for ECE, EEE, and EIE branches:

3. Figure of Merit of a Galvanometer. (CO4)
4. LCR circuit – Study of Resonance. (CO4)
5. Variation of Magnetic field along the axis of current – carrying circular coil (CO3)
6. Wedge method – Measurement of thickness of a foil. (CO1)
7. Specific charge (e/m) of an electron – J. J. Thomson method (CO3)
8. B-H Curve Unit- Determination of hysteresis loss (CO3)
9. Diffraction grating – Wavelength of laser light (CO1,CO5)
10. Photocell – Study of V – I characteristics, determination of work function. (CO3)
11. Optical Fiber – Determination of Numerical Aperture (CO5)
12. Torsional pendulum-Measurement of Rigidity Modulus (CO4)

D. Experiments for CE and ME branches:

3. AC Sonometer – Verification of vibrating laws (CO4)
4. Wedge method – Measurement of thickness of a foil (CO1)
5. Diffraction grating – Wavelength of laser light. (CO1, CO5)
6. Photocell – Study of V – I characteristics and determination of work function (CO3)
7. Torsional pendulum-Measurement of Rigidity Modulus (CO4)
8. Determination of Dielectric constant of a sample (CO2, CO3)
9. Optical Fiber – Determination of Numerical Aperture (CO5)
10. Compound pendulum – Measurement of ‘g’ (CO4)
11. Variation of Magnetic field along the axis of current – carrying circular coil (CO3)
12. Figure of Merit of a Galvanometer. (CO4)

Text Books:

1. Panigrahi, S., & Mallick, B. (2015), Engineering Practical Physics (1st ed.). Cengage Learning.
2. Madhusudhana Rao, C. V., & Vasanth Kumar, V. (2015), Engineering Physics Lab Manual (4th ed.). Scitech publications.
3. Ramarao Sri, Choudary Nityanand and Prasad Daruka, "Lab Manual of Engineering Physics"., Vth ed., Excell Books, 2010.

Virtual Lab References:

1. [1] <http://vlab.amrita.edu/?sub=1&brch=201&sim=366&cnt=1>
2. [2] <http://vlab.amrita.edu/?sub=1&brch=195&sim=840&cnt=1>
3. [3] <http://vlab.amrita.edu/?sub=1&brch=282&sim=879&cnt=1>

Web Resources:

1. Web Resources:
2. <http://plato.stanford.edu/entries/physics-experiment>
3. <http://www.physicsclassroom.com/The-Laboratory>
4. <http://facstaff.cbu.edu/~jvarrian/physlabs.html>

24UC182
AI TOOLS & APPLICATIONS
(CE/ECE/EEE/EIE/ME)

Course Category:	Engineering Science	Credits:	1
Course Type:	Practical	Lecture-Tutorial-Practice:	0-0-2
Pre-requisites:	Nil	Continuous Assessment: Summative Assessment: Total Marks:	60 40 100

Course Description:

This course introduces fundamental concepts and practical tools of Artificial Intelligence relevant to all engineering disciplines. It emphasizes the effective and ethical use of AI for productivity, analytical reasoning, research, and real-world engineering applications.

Course Objectives:

1. To provide an overview of AI and its applications in various engineering fields.
2. To demonstrate the use of AI as a personal assistant for productivity enhancement.
3. To explore AI as a tool for logical reasoning and mathematical analytics.
4. To utilize generative AI for literature review, gap analysis, ideation, and article writing.
5. To discuss detailed applications of AI in CE, Electronics, and ME.

Course Outcomes:

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

CO	Course Outcomes	BTL
CO1	Explain the basic concepts of AI and its historical evolution	K2
CO2	Apply AI tools for productivity enhancement	K3
CO3	Develop logical reasoning and analytical skills using AI tools	K3
CO4	Conduct literature reviews and write articles using generative AI	K3
CO5	Implement AI solutions in their respective engineering branches	K3

24UC101			
Essence of Indian Knowledge Tradition			
Course Category:	Mandatory (MC)	Credits:	0
Course Type:	Theory	Lecture-Tutorial-Practice:	2-0-0
Pre-requisites:	-	Continuous Assessment:	100
		Summative Assessment:	-
		Total Marks:	100
Course Description:			
<p>This course provides an in-depth exploration of the rich Indian knowledge tradition spanning philosophy, scientific advancements, cultural perspectives, educational heritage, and contemporary relevance. Students will delve into foundational texts, philosophical schools, scientific contributions, cultural artefacts, and the impact of Indian knowledge systems on global thought.</p>			
Course Objectives:			
<ol style="list-style-type: none"> 1. To introduce students to the major philosophical schools (Darshanas) of India and their foundational principles. 2. To familiarize students with significant scientific contributions in mathematics, astronomy, medicine, and technology from ancient India. 3. To explore the cultural perspectives of India through its literature, arts, architecture, and political systems. 4. To understand the educational heritage of ancient India and its impact on holistic learning and societal evolution. 5. To analyse the influence and contemporary relevance of Indian knowledge systems globally. 			
Course Outcomes:			
At the end of the course, the students will be able to			
CO	Course Outcomes		BTL
CO1	Analyse and differentiate between the various philosophical schools of India		K4
CO2	Assess the contributions of ancient Indian scientists and their impact on global knowledge systems.		K4
CO3	Interpret and discuss the cultural and societal norms depicted in Indian literature, arts, and architecture.		K3
CO4	Assess critically the educational methods and institutions of ancient		K4

	India and their relevance today.	
CO5	Identify the challenges and opportunities in reviving and adapting traditional Indian knowledge systems in contemporary contexts.	K3

Course articulation matrix

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

(1-Low, 2 -Medium, 3–High) (WK9)

Course Content

Unit-I

Foundations of Indian Philosophy

Introduction to Darshanas; detailed study of Nyaya, Vaisheshika, Sankhya, Yoga, Mimamsa and Vedanta.

Unit-II

Scientific Advancements

Mathematical theorems of Aryabhata, Astronomical discoveries of Brahmagupta, Principles of Ayurveda, Techniques in Metallurgy and Chemistry.

Unit-III

Cultural Perspectives

Analysis of Ramayana and Mahabharata, Study of classical texts (Puranas and Shastras), Evolution of temple architecture, Ethical governance principles.

Unit-IV

Educational Heritage of Ancient India

Gurukul system, Nalanda and Takshashila Universities, Role of women scholars, Influence of Southeast Asian education.

Unit-V

Influence and Contemporary Relevance

Spread of Indian philosophies to the West, Modern adaptations of Ayurveda, Challenges in preserving oral traditions.

Text Books:

1. Radhakrishnan, S., & Moore, C.A. (Eds.). (1957) *A Source Book in Indian Philosophy*. Princeton University Press.

Reference Books:

1. Basham, A. L. (1954). *The Wonder That Was India: A Survey of the Culture of the Indian Sub-Continent Before the Coming of the Muslims*. Sidgwick & Jackson.

Web Resources:

1. <https://iep.utm.edu/category/traditions/indian/>
2. <https://www.britannica.com/biography/Aryabhata-I>
3. <https://sacred-texts.com/hin/dutt/index.htm>
4. <https://nalandauniv.edu.in/>
5. <https://www.vidhyanjaliacademy.com/gurukul-education-system-in-ancient-india/>
6. <https://brownhistory.substack.com/p/how-indian-philosophies-influenced>