TECHNO INDIA UNIVERSITY

WEST BENGAL

Department of Computer Science and Engineering

Syllabus

for

4-Years B.Tech.

in

Computer Science and Engineering (CSE)

CURRICULUM

First Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-BS-UMA-T11101	Mathematics – I	3	1	0	4
2	TIU-BS-UPH-T11101	Physics	3	1	0	4
3	TIU-ES-UCS-T11101	Introduction to Programming	3	0	0	3
PRACTICAL						
1	TIU-BS-UPH-L11101	Physics Lab	0	0	3	1.5
2	TIU-ES-UCS-L11101	Introduction to Programming Lab	0	0	3	1.5
3	TIU-ES-UME-L11191	Engineering Drawing & Graphics	0	0	3	1.5
4	TIU-ES-UCS-L11191	Basic Computing Lab	0	0	2	1
SESSIONAL						
1	TIU-HSM-UES-S11191	Entrepreneurship Skill Development	0	0	2	1
2	TIU-HSM-UEN-S11191	Career Advancement & Skill Development - I Communication Skill	2	0	0	2
TOTAL CREDIT						**19.5**

Second Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-BS-UMA-T12101A	Mathematics-IIA	3	1	0	4
2	TIU-BS-UMA-T12102	Mathematics for Data Science	3	0	0	3
3	TIU-ES-UME-T12101	Engineering Mechanics	3	0	0	3
4	TIU-ES-UCS-T12101	Problem Solving Using Data Structures	3	0	0	3
5	TIU-ES-UEE-T12101	Basic Electrical & Electronics Engineering	3	1	0	4
PRACTICAL						
1	TIU-ES-UCS-L12101	Problem Solving using Data Structures Lab	0	0	3	1.5
2	TIU-ES-UCS-L12101	Basic Electrical & Electronics Engineering Lab & Simulation	0	0	3	1.5
SESSIONAL						
1	TIU-HSM-UES-S12191	Entrepreneurship Skill Development	0	0	2	1
2	TIU-HSM-UEN-S12191	Career Advancement & Skill Development - II Communication Skill	0	0	2	1
TOTAL CREDIT						**22**

Third Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-UMA-T215	Mathematics-III (Discrete Mathematics)	3	0	0	3
2	TIU-UMB-T201	Environmental Science	2	0	0	0
3	TIU-UEC-T211	Digital Electronics	3	0	0	3
4	TIU-UCS-T201	Data Structure and Algorithms	3	0	0	3
5	TIU-UCS-T207	Computer Organization	3	0	0	3
PRACTICAL						
1	TIU-UCS-L207	Computer Organization Lab	0	0	3	1.5
2	TIU-UEC-L211	Digital Electronics Lab	0	0	3	1.5
3	TIU-UCS-L209	Data Structure and Algorithms Lab	0	0	5	2.5
SESSIONAL						
1	TIU-UES-S281	Entrepreneurship Skill Development	0	0	2	1
CASD-III (Elective)						
1	TIU-UCS-S201A	Future of Work	0	1	2	2
2	TIU-UEN-S297	French	0	1	2	2
TOTAL CREDIT						**21.5**

Fourth Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-UMA-T206	Mathematics-IV	4	0	0	4
2	TIU-UCS-T220	Computer Architecture	3	0	0	3
3	TIU-UCS-T224	Graph Theory and Combinatorics	3	0	0	3
4	TIU-UCS-T214	Object Oriented Programming	3	0	0	3
5	TIU-UEC-T210	Microprocessor and Microcontroller	3	0	0	3
PRACTICAL						
1	TIU-UCS-L216	Numerical Lab	0	0	3	1.5
2	TIU-UCS-L220	Computer Architecture Lab	0	0	3	1.5
3	TIU-UCS-L214	Object Oriented Programming Lab	0	0	3	1.5
4	TIU-UEC-L218	Microprocessor and Microcontroller Lab	0	0	3	1.5
SESSIONAL						
1	TIU-UES-S282	Entrepreneurship Skill Development	0	0	2	1
CASD-IV						

1		Career Advancement & Skill Development-IV-Prompt Engineering	0	0	2	1
TOTAL CREDIT						**24**

Fifth Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-UCS-T321	Design and Analysis of Algorithms	3	0	0	3
2	TIU-UCS-T301	Database Management System	3	0	0	3
3	TIU-UCS-T317	Operating Systems	3	0	0	3
4	TIU-UCS-T323	Automata Theory	3	0	0	3
PRACTICAL						
1	TIU-UCS-L315	Database Management System Lab	0	0	3	1.5
2	TIU-UCS-L321	Design and Analysis of Algorithms Lab	0	0	3	1.5
3	TIU-UCS-L319	Object Oriented Systems Lab	0	1	2	2
4	TIU-UCS-L317	Operating Systems Lab	0	0	3	1.5
SESSIONAL						
1	TIU-UES-S381	Entrepreneurship Skill Development	0	0	1	1
					TOTAL CREDIT	**22.5**
CASD: Elective I						
1	TIU-UCS-S301A	Mobile Computing	3	0	0	3
2	TIU-UCS-S301B	Cryptography and Network Security	3	0	0	3
3	TIU-UCS-S301C	Artificial Intelligence	3	0	0	3
4	TIU-UCS-S301D	Generative AI	3	0	0	3
5	TIU-UCS-S301E	Information and Coding Theory	3	0	0	3
6	TIU-UCS-S301F	SAP	3	0	0	3

Sixth Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-UMA-T302	Operations Research & Optimization Techniques	3	0	0	3
2	TIU-UCS-T304	Computer Networks	3	0	0	3
3	TIU-UCS-T320	Compiler Design	3	0	0	3
4	TIU-UCS-T314	Software Engineering	3	0	0	3
5	TIU-UCS-E32#	Elective-I	3	0	0	3
PRACTICAL						
1	TIU-UCS-L394	Computer Networks Lab	0	0	3	1.5
2	TIU-UCS-L396	Compiler Design Lab	0	0	3	1.5
3	TIU-UCS-L384	Software Engineering Lab	0	0	3	1.5

4	TIU-UCS-L3##	Elective-I Lab	0	0	3	1.5
SESSIONAL						
1	TIU-UES-S398	Entrepreneurship Skill Development	0	0	1	1
2	TIU-UTR-S304	Career Advancement & Skill Development-VI Elective-III: Aptitude and Soft Skill	0	0	2	1
TOTAL CREDIT						**23**
ELECTIVE-I						
1	TIU-UCS-E330A	Image Processing	3	0	0	3
2	TIU-UCS-E322	Web Technology	3	0	0	3
3	TIU-UCS-E328	Computer Graphics	3	0	0	3
4	TIU-UCS-E326	Data Analytics	3	0	0	3
ELECTIVE-I Lab						
1	TIU-UCS-L330A	Image Processing Lab	0	0	3	1.5
2	TIU-UCS-L330B	Web Technology Lab	0	0	3	1.5
3	TIU-UCS-L330C	Computer Graphics Lab	0	0	3	1.5
4	TIU-UCS-L330D	Data Analytics Lab	0	0	3	1.5

Seventh semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
1	TIU-UCS-S403	Career Advancement & Skill Development-VII-Managerial Economics	2	0	0	2
2	TIU-UCS-E40#	Elective -II	3	0	0	3
3	TIU-UEC-E4##	Elective-III	3	0	0	3
1	TIU-UCS-L40#	Elective -II Lab	0	0	3	1.5
2	TIU-UCS-P495	Project -I	0	2	4	4
1	TIU-UES-S499	Entrepreneurship Skill Development	0	0	2	1
TOTAL CREDIT						**17.5**
Elective-II						
1	TIU-UCS-E419C	Machine Learning	3	0	0	3
2	TIU-UCS-E419B	Cloud Computing and IoT	3	0	0	3
3	TIU-UCS-E419D	Soft Computing	3	0	0	3
Elective-II Lab						

1	TIU-UCS-L419C	Machine Learning Lab	0	0	3	1.5
2	TIU-UCS-L419B	Cloud Computing and IoT Lab	0	0	3	1.5
3	TIU-UCS-L419D	Soft Computing Lab	0	0	3	1.5
Open Elective-III						
1	TIU-UCS-E415	Circuit Theory	3	0	0	3
2	TIU-UEC-E407	Digital Signal Processing	3	0	0	3
3	TIU-UEC-E409	VLSI Design	3	0	0	3
4	TIU-UEC-E411	Communication System	3	0	0	3

Eighth Semester

S. No	Course Code	Course Title	Contact Hrs. / Week			Credit
			L	T	P	
THEORY						
1	TIU-UMG-S400	Career Advancement & Skill Development-VIII-Values and Ethics	2	0	0	2
2	TIU-UCS-E40#	Elective-IV	3	0	0	3
PRACTICAL						
1	TIU-UCS-D498	Project-II (Final Thesis / Dissertation)	0	4	8	8
2	TIU-UCS-G496	Grand Viva	0	0	0	3
SESSIONAL						
1	TIU-UES-S498	Entrepreneurship Skill Development	0	0	2	1
TOTAL CREDIT						**17**
Elective-IV						
1	TIU-UCS-E416	Data Warehousing and Data Mining	3	0	0	3

2	TIU-UCS-E418	Natural Language Processing (NLP) and its applications	3	0	0	3
3	TIU-UCS-E420	Bioinformatics	3	0	0	3
4	TIU-UCS-E422	Distributed Operating System	3	0	0	3

DETAIL SYLLABUS

SEMSTER 1

TECHNO INDIA UNIVERSITY
WEST BENGAL

Department of Computer Science and Engineering

Career Advancement & Skill Development -in Communication Skill (TIU-HSM-UEN-S11191)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 1st Sem.
Course Title: Career Advancement & Skill Development - in Communication Skill	**Subject Code:** TIU-HSM-UEN-S11191
Contact Hours/Week: 2-0-0 (L-T-P)	**Credit: 2**

COURSE OBJECTIVE:

Enable the student to:

1. Develop English proficiency for clear, precise, and confident workplace communication.
2. Enhance practical skills in vocabulary, grammar, pronunciation, speaking, and writing.
3. Apply communication theories to improve professional and interpersonal interactions.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Explain fundamental communication principles and their relevance in workplace interactions.	K2
CO-2:	Apply grammar and language skills to construct precise and coherent spoken and written communication.	K3
CO-3:	Demonstrate fluency in spoken English through pronunciation drills, vocabulary building, and interactive conversations.	K4
CO-4:	Construct well-organized sentences, paragraphs, and linked paragraphs to enhance professional writing	K3
CO-5:	Develop and revise written communication by employing strategies for drafting, editing, and proofreading.	K3
CO-6:	Assess and refine communication skills to ensure clarity, precision, and confidence in workplace interactions.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO COMMUNICATION**	**7 Hours**
Definition of Communication, Importance of Communication in the Workplace, Introduction to Communication Theory, Elements of Effective Communication, Barriers to Communication, Verbal and Non-Verbal Communication, Role of Culture in Communication.		
MODULE 2:	**LANGUAGE AND GRAMMAR SKILLS**	**5 Hours**
Fundamentals of English Grammar, Sentence Structure and Syntax, Parts of Speech, Tenses and their Usage, Common Errors in Grammar, Punctuation and Mechanics, Effective Use of Vocabulary, Word Formation and Usage, Formal vs. Informal Language.		
MODULE 3:	**SPEAKING SKILLS**	**8 Hours**
Principles of Effective Speaking, Pronunciation Drills, Sounds of English: Vowels and Consonants, Stress and Intonation, Developing Conversational Skills, Speaking with Clarity and Confidence, Public Speaking Basics, Expressing Opinions and Arguments, Active Listening and Response.		
MODULE 4:	**WRITING SKILLS**	**8 Hours**
The Writing Process: Planning, Drafting, Revising, Editing, Writing Effective Sentences and Paragraphs, Paragraph Development and Coherence, Formal and Informal Writing Styles, Writing Emails and Workplace Documents, Writing Reports and Memos, Common Writing Errors and How to Avoid Them		
MODULE 5:	**PRACTICAL LANGUAGE APPLICATION**	**5 Hours**
Building Vocabulary through Context, Word Choice and Precision, Constructing Grammatically Correct Sentences, Exercises in Sentence Formation, Pronunciation Drills and Accent Neutralization, Role-Plays and Dialogues, Group Discussions and Debates, Writing and Structuring Paragraphs, Linking Paragraphs for Coherent Writing.		
MODULE 6:	**PROFESSIONAL COMMUNICATION IN THE WORKPLACE**	**4 Hours**
Workplace Communication Etiquette, Business Correspondence, Writing Professional Emails, Preparing Presentations, Communicating in Meetings, Handling Workplace Conversations, Persuasive and Negotiation Skills, Overcoming Communication Barriers, Strategies for Effective Workplace Communication.		
TOTAL LECTURES		**30 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	3	-	-	2	3	-	2	-	-	-
CO2	-	-	-	-	-	-	-	-	-	3	-	2	-	-	-
CO3	-	-	-	-	-	-	-	-	-	3	-	2	-	-	-
CO4	-	-	-	-	-	-	-	-	-	3	2	2	-	-	-
CO5	-	-	-	-	-	-	-	-	-	3	2	2	-	-	-
CO6	-	-	-	-	-	3	-	-	3	3	-	3	-	-	-
						3			2.5	3	2	2.16			

Books:

1. Sanjay Kumar, Pushp Lata, "Communication Skills", Oxford University Press, 2015, ISBN: 9780199457069
2. M Ashraf Rizvi, "Effective Technical Communication", McGraw Hill Education, 2017, ISBN9352606108
3. Steven A. Beebe, Susan J. Beebe, and Mark V. Redmond, "Interpersonal Communication: Relating to Others", Pearson, 2013, ISBN-10: 020586273X, ISBN-13: 978-0205862733.
4. Judee K. Burgoon, Laura K. Guerrero, and Kory Floyd, "Nonverbal Communication", Routledge, 2016, ISBN-10: 1138121348, ISBN-13: 978-1138121346.
5. Ronald B. Adler, Lawrence B. Rosenfeld, and Russell F. Proctor II, "Interplay: The Process of Interpersonal Communication", Oxford University Press, 2017, ISBN-10: 019064625X, ISBN-13: 978-0190646257.
6. Joseph A. DeVito, "The Interpersonal Communication Book", Pearson, 2015, ISBN-10: 0133753816, ISBN-13: 978-0133753813.
7. Sarah Trenholm and Arthur Jensen, "Interpersonal Communication", Oxford University Press, 2013, ISBN-10: 0199827504, ISBN-13: 978-0199827503.
8. John Stewart, "Bridges Not Walls: A Book About Interpersonal Communication", McGraw-Hill Education, 2011, ISBN-10: 0073534315, ISBN-13: 978-0073534312.
9. Pamela J. Kalbfleisch, "Interpersonal Communication: Evolving Interpersonal Relationships", Routledge, 2013, ISBN-10: 0805816611, ISBN-13: 978-0805816619.
10. Mark L. Knapp, John A. Daly, and Frederick P. M. Boster, "Interpersonal Communication Handbook", Sage Publications, 2011, ISBN-10: 1412974747, ISBN-13: 978-1412974745

Mathematics-I (TIU-BS-UMA-T11101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 1st Sem.
Course Title: Mathematics-I	**Subject Code:** TIU-BS-UMA-T11101
Contact Hours/Week: 3-1-0 (L-T-P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. Analyze and describe the behavior of functions of single and multiple variables, understand sequences and series.
2. Solve systems of linear equations, evaluate eigen values and eigenvectors of square matrices.
3. Analyzing differential equations and finding their solutions.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Analyze the behavior and the nature of the curve with calculus of one variable.	K4
CO-2:	Develop a basic understanding of functions of several variables and their properties.	K4
CO-3:	Investigate the solutions of systems of linear equations using Determinants and Matrices.	K4
CO-4:	Evaluate Eigenvalue and vectors of square matrices.	K4
CO-5:	Examine the nature (viz., convergence, divergence) of sequence and series.	K4
CO-6:	Analyze differential equations and investigate solutions.	K4

COURSE CONTENT:

MODULE 1:	**Differential Calculus**	**12 Hours**
Differential Calculus (Functions of one variable): Rolle's theorem (statement only), Cauchy's mean value theorem (Lagrange's mean value theorem as a special case), Taylor's and Maclaurin's theorems with remainders, indeterminate forms, concavity and convexity of a curve, points of inflexion, asymptotes and curvature. Differential Calculus (Functions of several variables): Limit, continuity and differentiability of functions of several variables, partial derivatives and their geometrical interpretation, differentials, derivatives of composite and implicit functions, derivatives of higher order and their commutatively, Euler's theorem on homogeneous functions, harmonic functions, Taylor's expansion of functions of several variables, maxima and minima of functions of several variables – Lagrange's method of multipliers.		
MODULE 2:	**Ordinary Differential Equations**	**10 Hours**
Ordinary Differential Equations: Formation of differential equations, First order differential		

equations - exact, linear and Bernoulli s form, second order differential equations with constant coefficients, method of variation of parameters, general linear differential equations with constant coefficients, Euler's equations, system of differential equations.		
MODULE 3:	**Sequences and Series**	**8 Hours**
Sequences and Series: Sequences and their limits, convergence of series, comparison test, Ratio test, Root test, Absolute and conditional convergence, alternating series, Power series.		
MODULE 4:	**Matrix and Determinant**	**15 Hours**
Matrix and Determinant: Revision of matrix and determinant, rank and nullity, solutions of system of linear equations using Determinants and Matrices; Eigenvalues and eigen vectors, Cayley-Hamilton Theorem, transformation of matrices, adjoint of an operator, normal, unitary, hermitian and skew-hermitian operators, quadratic forms.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	2	-	-	-	-	-	-	-	-	2	3	-	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	3	-	-
CO5	3	2	2	-	-	-	-	-	-	-	-	2	3	-	-
CO6	3	2	2	-	-	-	-	-	-	-	-	3	3	-	-
	3	2.2	2.25									2.16	3		

Books:

1. Higher Engineering Mathematics, B. S. Grewal
2. Advanced Engineering Mathematics, Kreyszig
3. A TextBook of Engineering Mathematics, Rajesh Pandey
4. Engineering Mathematics, B. K. Pal, K. Das

Introduction to Programming (TIU-ES-UCS-T11101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 1st Sem.
Course Title: Introduction to Programming	**Subject Code:** TIU-ES-UCS-T11101
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

Enable the student to:

1. Develop algorithmic problem-solving skills and implement them in C programs.
2. Apply modular programming, recursion, and data structures to create interactive C programs.
3. Utilize advanced C concepts like structures, pointers, and linked lists for efficient programming.

COURSE OUTCOME:

The students will be able to:

CO1:	Analyze algorithmic solutions to problems.	K4
CO2:	Construct algorithms using C programming.	K3
CO3:	Apply interactive input/output, arithmetic expressions, repetitions, decision-making, and arrays in programs.	K3
CO4:	Organize modular C programs using functions, including recursion.	K3
CO5:	Categorize programs using structures, unions, pointers, and linked lists.	K4
CO6:	Utilize file input and output operations in programs.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO C LANGUAGE**	**4 Hours**
Character set, Variables and Identifiers, Built-in Data Types, Variable Definition, Arithmetic operators and Expressions, Constants and Literals, Simple assignment statement, Basic input/output statement, Simple 'C' programs.		
MODULE 2:	**CONDITIONAL STATEMENTS AND LOOPS**	**6 Hours**
Decision making within a program Conditions, Relational Operators, Logical Connectives, if statement, if-else statement. Loops: while loop, do while, for loop, Nested loops, Infinite loops, switch statement, Structured Programming.		
MODULE 3:	**ARRAYS**	**6 Hours**
One dimensional array: Array manipulation, Searching, Insertion, and Deletion of an element from an array, finding the largest / smallest element in an array; Two dimensional arrays, Addition/ multiplication of two matrices transpose of a square matrix, Null terminated strings as array of characters, Representation sparse matrice.		
MODULE 4:	**FUNCTIONS**	**7 Hours**
Top-down approach of problem solving; Modular programming and functions; Standard Library of		

C functions; Prototype of a function Formal parameter list, Return Type, Function call, Block structure; Passing arguments to a Function Call by reference, Call by value, Recursive Functions, Arrays as function arguments.		
MODULE 5:	**STRUCTURES AND UNIONS**	**5 Hours**
Structure variables, Initialization, Structure assignment, Nested structure, Structures and Functions, Structures and arrays: Arrays of structures, Structures containing arrays, Unions.		
MODULE 6:	**POINTERS**	**9 Hours**
Address operators, Pointers type declaration, Pointer assignment, Pointer initialization, Pointer arithmetic, Functions and pointers, Arrays and Pointers, Pointer arrays.		
MODULE 7:	**SELF-REFERENTIAL STRUCTURES AND LINKED LISTS**	**3 Hours**
Creation of a singly connected linked list, traversing a linked list, Insertion into a linked list, Deletion from a linked list.		
MODULE 8:	**FILE PROCESSING**	**5 Hours**
Concept of Files, File opening in various modes and closing of a file, Reading from a file, writing onto a file.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	2	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO3	2	-	2	-	3	-	-	-	-	-	-	-	3	-	-
CO4	2	-	2	-	3	-	-	-	-	-	-	-	-	-	3
CO5	2	2	3	-	2	-	-	-	-	-	-	-	-	-	3
CO6	-	-	-	-	3	-	-	-	-	2	-	-	-	-	3
	2.2	2.3	2.333		2.6					2			2.667		3

Books:

1. B W Kernighan and D.M. Ritchie, The C Programming Language, Prentice Hall of India.
2. K. Venugopal and Sudeep R Prasad, Programming with C, McGraw Hill
3. R G Dromey, How to solve it by Computer, Prentice Hall in India.
4. Jones, Robin and Stewart, The Art of C Programming, Narosa Publishing House
5. A Kenneth, C Problem solving and Programming, Prentice Hall International.
6. H.Scheldt, C: The Complete Reference, 4th Edition, McGraw Hill

Physics (TIU-BS-UPH-T11101)

Program: BTech in CSE	**Year, Semester:** 1st Yr., 1st Sem
Course Title: Physics	**Subject Code:** TIU-BS-UPH-T11101
Contact Hours/Week: 3–1–0 (L–T–P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. Provide a foundational understanding of basic concepts of physics.
2. Develop problem-solving skills and apply the basic concepts of physics in real-world phenomena.
3. Foster critical thinking and analytical skills in applying theoretical knowledge to practical physics problems.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Explain and apply the concepts of gradient, divergence, and curl in vector calculus and their applications in mechanics.	K3
CO-2:	Apply the principles of simple harmonic motion, damped and forced oscillations and resonance to real-world acoustical problems.	K3
CO-3:	Describe the principles of interference, diffraction, and polarization, and explain the working of lasers, including population inversion and pumping mechanisms. Also they will be able to state and explain Maxwell's equations and their implications for electromagnetic wave propagation.	K2
CO-4:	Understand and explain the basic principles of wave mechanics, including the Schrödinger equation, probability interpretation, and quantum harmonic oscillators.	K2
CO-5:	Define the key concepts of phase space, macrostates, microstates, and describe the statistical distributions like Maxwell-Boltzmann, Fermi-Dirac, and Bose-Einstein.	K2
CO-6:	Understand the fundamental concepts in solid-state physics (e.g., crystal structure, magnetization, and hysteresis) and thermodynamics (e.g., black body radiation, entropy, and the laws of thermodynamics).	K2

COURSE CONTENT:

MODULE 1:	**MECHANICS**	**6 Hours**
Vector Calculus- gradient, divergence, curl; Frame of references, Mechanics of a single particle - conservative and non-conservative forces, potential energy function F= -grad V		
MODULE 2:	**ACCOUSTICS**	**5 Hours**
Simple harmonic oscillator, damped and forced motion and resonance; wave motion and equation.		

MODULE 3:	OPTICS	7 Hours
Interference - overview of interference phenomena, interference due to thin films- Newton's ring; Diffraction - single slit, double slit and grating; Polarization: introduction, polarization by reflection, scattering of light, circular and elliptical polarization, optical activity; Lasers - principle and working of laser, population inversion, pumping, various modes, threshold population inversion with examples.		
MODULE 4:	**ELECTROMAGNETISM**	**4 Hours**
Introduction (qualitative discussion), Maxwell's equations, wave equation, plane electromagnetic waves, Poynting's theorem.		
MODULE 5:	**WAVE MECHANICS**	**7 Hours**
Introduction to quantum physics, wave functions and Schrodinger equation, probability interpretation, particle in a 1D box, quantum harmonic oscillator, Hydrogen atom problem.		
MODULE 6:	**STATISTICAL MECHANICS**	**6 Hours**
Qualitative ideas about phase space, macrostates and microstates, density of states, qualitative discussion on Maxwell-Boltzmann, Fermi-Dirac and Bose-Einstein Statistics.		
MODULE 7:	**SOLID STATE PHYSICS**	**6 Hours**
Introduction of crystal structure, Bragg's law; Properties and applications of dielectric materials, Magnetisation- permeability and susceptibility, classification of magnetic materials, ferromagnetism, magnetic domains and hysteresis, applications.		
MODULE 8:	**THERMAL PHYSICS**	**3 Hours**
Black body radiation, 1st and 2nd law of thermodynamics, concept of entropy.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	3	2	-	-	-	-	-	-	-	-	2	3	-	-
CO3	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO4	3	2	-	-	-	-	-	-	-	-	-	3	3	-	-
CO5	3	2	-	-	-	-	-	-	-	-	-	3	3	-	-
CO6	3	2	-	-	-	-	-	-	-	-	-	3	3	-	-
	3	2.16	2									2.5	3		

Books:

1. Introduction to Electrodynamics, David J. Griffiths, Pearson Education India Learning Private Limited
2. Introduction to Classical Mechanics, R Takwale, P Puranik, McGraw Hill Education private limited
3. Engineering Physics , Dattuprasad Ramanlal Joshi, McGraw Hill Education private limited

4. Quantum Physics of Atoms, Molecules, Solids, Nuclei and Particles,Robert Eisberg, Robert Resnick, Wiley
5. Statistical Physics, L.D. Landau, E M.Lifshitz, Butterworth-Heinemann
6. Optics,Ghatak, McGrawHill Education India Private Limited
7. Engineering Physics , Hitendra K Malik & A K Sing, McGraw Hill Education private limited
8. Advanced Acoustics, Dr. D.P. Raychaudhuri, The new bookstall, Revised Ninth Edition, 2009
9. Concepts of Modern Physics (Sixth Edition) by Arthur Beiser (Published by McGraw-Hill).
10. Introduction to Solid State Physics (January2019) by Charles Kittel (Published by Wiley)

Physics Lab (TIU-BS-UPH-L11101)

Program: B.Tech in CSE	**Year, Semester:** 1st Yr, 1st Sem
Course Title: : Physics Lab	**Subject Code:** TIU-BS-UPH-L11101
Contact Hours/Week: 0–0–3(L–T–P)	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Provide hands-on experience with experimental techniques in optics, electricity, and mechanics
2. Develop a strong understanding of the fundamental physical constants and properties of materials
3. Enhance students' problem-solving and analytical skills through real-world applications

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Develop hands-on skills in setting up experimental apparatus and accurately measuring physical quantities.	K3
CO-2:	Analyze experimental data using appropriate methods, interpret results, and assess the reliability and accuracy of measurements.	K4
CO-3:	Correlate theoretical physics principles with experimental observations to understand real-world applications.	K5
CO-4:	Demonstrate the ability to troubleshoot experimental issues and make informed decisions to optimize accuracy.	K5
CO-5:	Document experiments systematically and effectively present results, including calculations and error analysis.	K6
CO-6:	Work collaboratively in a lab environment, maintaining safety protocols and contributing to group discussions and analysis.	K6

COURSE CONTENT:

MODULE 1:	**EXPERIMENT: 1**	**3 Hours**
To determine the wavelength of a monochromatic light by Newton's ring		
MODULE 2:	**EXPERIMENT: 2**	**3 Hours**
To determine the dispersive power of a prism		

MODULE 3:	**EXPERIMENT: 3**	**3 Hours**
To determine the unknown resistance by Carey-Foster bridge		
MODULE 4:	**EXPERIMENT: 4**	**3 Hours**
Determination of e/m ratio of electron by J.J. Thomson method		
MODULE 5:	**EXPERIMENT: 5**	**3 Hours**
Determination of Plank's constant using photoelectric effect		
MODULE 6:	**EXPERIMENT: 6**	**3 Hours**
To determine the refractive index of water using travelling microscope		
MODULE 7:	**EXPERIMENT: 7**	**3 Hours**
To determine the Young's modulus/bending moment of a beam by flexure method		
MODULE 8:	**EXPERIMENT: 8**	**3 Hours**
To determine the rigidity modulus of a wire by dynamic method		
MODULE 9:	**EXPERIMENT: 9**	**3 Hours**
To determine the viscosity of water by capillary tube method		
MODULE 10:	**EXPERIMENT: 10**	**3 Hours**
To determine the thermal conductivity of a bad conductor by hot wire method		
MODULE 11:	**EXPERIMENT: 11**	**3 Hours**
Study of B-H loop of a magnetic material		
MODULE 12:	**EXPERIMENT: 12**	**3 Hours**
Study of electrical resistivity of metal and semiconductors by four probe method		
Total Hours (Any seven experiments to be performed)		**21 Hour**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	2	-	-	-	-	-	-	-	3	3	-	-
CO2	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	2	-	3	-	-	-	-	-	-	-	2	3	-	-
CO4	-	3	2	3	-	-	-	-	-	-	-	2	3	-	-
CO5	-	2	-	-	-	-	-	-	-	3	2	3	3	-	-

CO6	-	-	-	-	-	-	-	-	3	3	2	3	3	-	-
	3	2.5	2	2.5					3	3	2	2.5	3		

Books:

1. Laboratory Manual
2. Advanced Practical Physics (Volume I and II) for BSc Physics Lab, B. Ghosh & K.G Mazumdar
3. An advanced course in practical physics by D. Chattopadhyay and P.C Rakshit, New central agency(P)Ltd.

Introduction to Programming Lab (TIU-ES-UCS-L11101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 1st Sem
Course Title: Introduction to Programming Lab	**Subject Code:** TIU-ES-UCS-L11101
Contact Hours/Week: 0–0–3 (L–T–P)	**Credit:** 1.5

COURSE OBJECTIVE

Enable the student to:

1. Introduce students to the fundamentals of C programming, including syntax, data types, operators, and control structures, enabling them to write and execute basic programs.
2. Develop students' ability to analyze problems, apply algorithmic thinking, and implement solutions using decision-making constructs, loops, functions, and data structures.
3. Equip students with hands-on experience in using arrays, strings, pointers, structures, and unions, enabling them to develop efficient programs for mathematical computations, data processing, and real-world applications.

COURSE OUTCOME

CO-1	Demonstrate the ability to write, compile, and execute simple C programs using basic input-output functions, arithmetic operations, and control statements.	K2
CO-2	Apply conditional statements (if-else, ternary operator, switch-case) and looping constructs (for, while, do-while) to solve mathematical and logical problems.	K3
CO-3	Solve mathematical problems such as factorial, permutations & combinations, series summation, and trigonometric computations using C programming.	K3
CO-4	Develop programs using arrays and strings to perform operations such as searching, sorting, frequency analysis, and string transformations.	K4
CO-5	Utilize pointers, structures, and unions in C to perform complex operations such as matrix manipulations, complex number arithmetic, and data	K4

	organization.	
CO-6	Implement user-defined functions and demonstrate the ability to use memory management functions, pointers, and structures for efficient data handling.	K4

COURSE CONTENT

MODULE 1:	**Introduction to C Programming & Basic Operations**	**6 Hours**
Writing and executing a basic C program (Hello World). Understanding Input/Output functions (printf(), scanf()). Variables, Data Types, and Memory Allocation. Arithmetic operations and simple mathematical computations		
MODULE 2:	**Control Structures & Decision Making**	**6 Hours**
Conditional statements (if-else, ternary operator, switch-case). Looping constructs (for, while, do-while). Nested control structures.		
MODULE 3:	**Functions, Recursion & Pattern Printing**	**6 Hours**
Defining and calling user-defined functions. Function parameters, return types, and recursion. Printing patterns using loops (*, numbers, alternating 0/1). Mathematical computations using recursion (Factorial, nCr).		
MODULE 4:	**Arrays & Strings**	**9 Hours**
One-dimensional and two-dimensional arrays. Searching & sorting algorithms. String operations (length, frequency analysis, conversion to uppercase/lowercase).		
MODULE 5:	**Pointers, Structures & Memory Management**	**9 Hours**
Pointer concepts and memory addresses. Pointer arithmetic and array manipulation using pointers. Structures and Unions for data organization. Dynamic memory allocation concepts.		
MODULE 6:	**Advanced Programming & Applications**	**9 Hours**
Matrix operations (Addition, Multiplication). Trigonometric function computations (sin, cos values at intervals). File handling concepts (basic read/write operations).		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	2	-	2	-	3	-	-	-	-	-	-	-	3	-	-
CO2	3	2	-	-	3	-	-	-	-	-	-	-	3	-	-
CO3	3	2	-	-	2	-	-	-	-	-	-	-	2	-	-
CO4	2	-	2	-	3	-	-	-	-	-	-	-	-	-	3
CO5	2	-	2	-	3	-	-	-	-	-	-	-	-	-	3
CO6	2	-	2	-	3	-	-	-	-	-	-	-	-	-	3

	2.33	2	2		2.833								2.666		3

Books:

1. B W Kernighan and D.M. Ritchie, The C Programming Language, Prentice Hall of India.
2. K. Venugopal and Sudeep R Prasad, Programming with C, McGraw Hill
3. R G Dromey, How to solve it by Computer, Prentice Hall in India.

Engineering Drawing and Graphics (TIU-ES-UME-L11191)

Program: B. Tech in CSE	**Year, Semester:** 1st year, 1st Semester
Course Title: Engineering Drawing and Graphics	**Subject Code:** TIU-ES-UME-L11191
Contact hours/week: 0-0-3 (L-T-P)	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Develop an understanding of the fundamental concepts and significance of engineering drawing in various engineering disciplines.
2. Acquire skills to construct and analyze engineering curves, projections of points, lines, planes, and solids.
3. Learn to interpret and create orthographic and isometric projections using conventional and computer-aided drafting techniques.
4. Gain proficiency in using drafting software for preparing accurate engineering drawings.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Understand the fundamental principles and scope of engineering drawing across various engineering disciplines.	K2
CO-2:	Demonstrate proficiency in constructing and analyzing different engineering curves.	K3
CO-3:	Apply projection techniques for points, lines, planes, and solids in different orientations.	K3
CO-4:	Develop skills to create orthographic and isometric projections accurately.	K3
CO-5:	Interpret and convert between pictorial, orthographic, and isometric views of objects.	K3
CO-6:	Utilize computer-aided drafting tools to create precise engineering drawings.	K3

COURSE CONTENT:

MODULE 1:	**Introduction**	**6 Hours**
Scope of Engineering Drawing in all Branches of Engineering, Uses of Drawing Instruments and Accessories, Types of Arrowheads, Lines, Dimension System, Representative Fraction, Types of Scales (plain and Diagonal Scale).		

MODULE 2:	**Engineering Curves**	**6 Hours**
Classification of Engineering Curves, Application of Engineering Curves, Constructions of Engineering Curves (Conics-ellipse; parabola; hyperbola with Tangent and Normal).		
MODULE 3:	**Projection of Points and Straight Lines**	**9 Hours**
Types of Projections - Oblique, Perspective, Orthographic and Isometric Projections; Introduction to Principal Planes of Projections, Projections of Points located in all four Quadrants; Projections of lines inclined to one of the Reference Plane and inclined to two Reference Planes.		
MODULE 4:	**Projections of Planes and Solids**	**9 Hours**
Projections of various planes (Polygonal, Circular, Elliptical shape inclined to one of the reference planes and two of the reference planes) and Projections of Solids (cube, prism, pyramid, cylinder, cone and sphere).		
MODULE 5:	**Orthographic Projections & Isometric View/Projections**	**8 Hours**
Projections on Principal Planes from Front, Top and Sides of the Pictorial view of an Object, First Angle Projection and Third Angle Projection system; Full Sectional Orthographic Views, Conversion of Orthographic Views into Isometric Projection, View or Drawing; Isometric Scale.		
MODULE 6:	**Overview of Computer Aided Drafting Tools**	**1 Hours**
Introduction to Computer Aided Drafting Software; Basic Tools; Preparation of Orthographic Projections and Isometric Views Using Drafting Software.		
	TOTAL LAB HOURS	**39 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	2	3	-	-	-	-	-	-	-	-	2	3	-	-
CO4	3	-	3	-	-	-	-	-	-	-	-	2	3	-	-
CO5	3	-	2	-	-	-	-	-	-	3	-	2	3	-	-
CO6	3	-	2	-	3	-	-	-	-	3	-	3	3	-	-
	3	2.33	2.5	2	3					3		2.16	3		

Books:

Main Reading:

1. Jolhe, Dhananjay A, Engineering Drawing an introduction to AutoCAD, Tata McGraw-Hill.
2. N.D. Bhatt, Engineering Drawing, Charotar Publishing House Pvt. Ltd.

Basic Computing Lab (TIU-ES-UCS-L11191)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 1st Sem.
Course Title: Basic Computing Lab	**Subject Code:** TIU-ES-UCS-L11191
Contact Hours/Week: 0–0–2 (L–T–P)	**Credit:** 1

COURSE OBJECTIVE:

Enable the student to:

1. To introduce students to the UNIX/Linux environment and familiarize them with fundamental system operations, commands, and file management techniques.
2. To develop proficiency in shell scripting and command-line utilities for automating tasks, managing processes, and handling files efficiently.
3. To provide hands-on experience with GitHub operations and debugging techniques while enhancing students' ability to work with text processing tools, redirection, and file compression in a UNIX/Linux environment.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1	Be Familiar with the UNIX/Linux operating system	K2
CO-2	Develop proficiency in using shell commands and writing basic shell scripts.	K3
CO-3	Understand file systems, process management, and user permissions.	K2
CO-4	Understand basic github operations and debugging of programs	K3
CO-5	Apply fundamental text processing tools and commands such as grep, find, and text editors (vi/nano) for efficient file manipulation and searching.	K4
CO-6	Utilize redirection, piping, and file compression techniques to manage data effectively in a UNIX/Linux environment.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO UNIX/LINUX AND BASIC COMMANDS**	**9 Hours**
Overview of UNIX/Linux operating systems, Logging into UNIX/Linux systems, Basic system commands: ls, cd, pwd, cp, mv, rm, clear, man, who, date, cal, etc. Understanding the file system hierarchy: /, /home, /bin, /usr, /var, etc.		
MODULE 2:	**FILE AND PROCESS MANAGEMENT**	**9 Hours**
File and Directory Management: Creating, removing, and organizing files and directories, Commands: mkdir, rmdir, touch, chmod, chown, rm, find, etc. Understanding file permissions and ownership (rwx permissions, chmod command) Process Management: Viewing active processes (ps, top, htop), Controlling processes: kill, bg, fg, jobs, nice, and renice,Understanding process states: running, sleeping, zombie.		
MODULE 3:	**TEXT PROCESSING AND BASIC SHELL SCRIPTING**	**9 Hours**
Text Editors (vi, nano): Creating, editing, saving, and existing files, Working with commands like		

grep, cat, more, less, sed, and awk Basic Shell Scripting: Writing simple shell scripts (bash), Understanding variables, loops (for, while), and conditional statements (if, elif, else), Creating automation scripts for file operations and system monitoring		
MODULE 4:	**REDIRECTION, PIPING, AND FILE COMPRESSION**	**9 Hours**
Redirection and Piping: Input/output redirection (>, >>, <) Piping (|) for command chaining File Compression and Archiving: Working with gzip, tar, zip, unzip,Creating and extracting archives for data backup		
MODULE 5:	**GITHUB BASICS AND DEBUGGING TECHNIQUES**	**9 Hours**
Using GitHub for Version Control: Setting up a GitHub repository, Basic commands: git init, git add, git commit, git push, git pull, git clone,Checking in and checking out files Debugging Techniques: Identifying and resolving errors in shell scripts, Using debugging tools (echo, set -x, gdb for C programs)		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	3	-	-	-	-	-	-	2	3	-	-
CO2	2	-	2	-	3	-	-	-	-	-	-	-	3	-	-
CO3	2	-	-	-	3	-	-	2	-	-	-	-	3	-	-
CO4	-	-	2	-	3	-	-	-	-	2	-	-	-	2	3
CO5	2	-	-	-	3	-	-	-	-	-	-	-	3	-	-
CO6	2	-	-	-	3	-	-	-	-	-	-	-	-	-	3
	2		2		3			2		2		2	3	2	3

Books:

1. "UNIX and Linux System Administration Handbook" – Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, and Dan Mackin
2. "The Linux Command Line: A Complete Introduction" – William E. Shotts Jr.
3. "Learning the bash Shell"– Cameron Newham.

SEMESTER 2

CAREER ADVANCEMENT & SKILL DEVELOPMENT-II - COMMUNICATION SKILL (TIU-HSM-UEN-S12191)

Program: B.Tech in CSE	**Year, Semester:**1st Year, 2nd Sem
Course Title: CAREER ADVANCEMENT & SKILL DEVELOPMENT-II - COMMUNICATION SKILL	**Subject Code:** TIU-HSM-UEN-S12191
Contact Hours/Week: 0-0-2 (L-T-P)	**Credit:** 1

COURSE OBJECTIVE:

Enable the student to:

1. Develop fluency in spoken and written English for clear, precise, and confident communication.
2. Train in formal writing, reports, proposals, and multimedia presentations.
3. Strengthen people skills, time management, and analytical reading for workplace success.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Explain fundamental communication principles and assess their relevance in workplace interactions.	K2
CO-2:	Apply grammar and language skills to construct precise and coherent spoken and written communication	K3
CO-3:	Demonstrate fluency in spoken English through practicing pronunciation drills, developing vocabulary, and engaging in interactive conversations.	K4
CO-4:	Construct well-organized sentences and paragraphs to enhance professional writing.	K3
CO-5:	Develop and revise written communication by employing strategies for drafting, editing, and proofreading	K3
CO-6:	Assess and refine communication skills to ensure clarity, precision, and confidence in workplace interactions.	K4

COURSE CONTENT:

MODULE 1:	**COMMUNICATION THEORY AND WORKPLACE DYNAMICS**	**7 Hours**
Definition of Communication, Communication Models, Workplace Communication Strategies, Effective Messaging, Organizational Communication, Cultural Communication, Verbal and Non-Verbal Cues, Barriers to Communication, Interpersonal and Group Communication		
MODULE 2:	**ADVANCED LANGUAGE AND GRAMMAR PROFICIENCY**	**5 Hours**
Morphology and Syntax, Sentence Structuring, Advanced Grammar Rules, Tense Modulation, Phrasal Verbs, Modifiers, Cohesion and Coherence, Lexical Resource, Semantics, Formal vs. Informal Register		
MODULE 3:	**STRATEGIC SPEAKING AND ORAL PROFICIENCY**	**8 Hours**

Phonetics and Phonology, Pronunciation Refinement, Stress and Intonation, Articulation and Clarity, Persuasive Speaking, Argumentation and Debate, Spontaneous Speaking, Interview Techniques, Business Pitches, Active Listening Strategies		
MODULE 4:	**PROFESSIONAL AND TECHNICAL WRITING**	**8 Hours**
Writing Process Methodologies, Text Structuring, Precision in Writing, Report Writing, Business Proposals, Formal Correspondence, Executive Summaries, Editing and Proofreading, Technical Documentation, Press Releases, Persuasive and Analytical Writing		
MODULE 5:	**APPLIED LANGUAGE AND COMMUNICATION EXERCISES**	**5 Hours**
Lexical Expansion, Idiomatic Expressions, Context-Based Learning, Grammar in Context, Role-Plays and Simulations, Speech Analysis, Storytelling Techniques, Collaborative Writing, Dialogues, Workplace Case Studies		
MODULE 6:	**CORPORATE COMMUNICATION AND LEADERSHIP SKILLS**	**4 Hours**
Professional Etiquette, Negotiation Tactics, Conflict Resolution, Crisis Communication, Leadership and Persuasion, Presentation Design, Cross-Cultural Communication, Media and Public Relations, Digital Communication Ethics, High-Stakes Conversations		
	TOTAL LECTURES	**30 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	2	3	-	-	2	3	-	-
CO2	-	-	-	-	-	-	-	-	2	3	-	-	3	-	-
CO3	-	-	-	-	-	-	-	-	3	2	-	-	3	-	-
CO4	-	-	-	-	-	-	-	-	-	3	-	2	3	-	-
CO5	-	-	-	-	-	-	-	-	-	2	3	-	3	-	-
CO6	-	-	-	-	-	-	-	2	3	-	-	2	3	-	-
								2	2.75	2.5	3	2	3		

Books:

1. Sanjay Kumar, Pushp Lata, “Communication Skills”, Oxford University Press, 2015, ISBN: 9780199457069
2. M Ashraf Rizvi, “Effective Technical Communication”, McGraw Hill Education, 2017, ISBN 9352606108
3. Sarah Trenholm and Arthur Jensen, “Interpersonal Communication”, Oxford University Press, 2017, ISBN-10: 019064625X, ISBN-13: 978-0190646257
4. Claude G. Théoret, “Advanced Communication Skills: 7 Keys to Personal and Professional Growth”, Independently Published, 2020, ISBN-10: 1656945618, ISBN-13: 978-1656945615..

5. Ronald B. Adler, Lawrence B. Rosenfeld, and Russell F. Proctor II, "Interplay: The Process of Interpersonal Communication", Oxford University Press, 2017, ISBN-10: 019064625X, ISBN-13: 978-0190646257.
6. Joseph A. DeVito, "The Interpersonal Communication Book", Pearson, 2015, ISBN-10: 0133753816, ISBN-13: 978-0133753813.
7. Mark L. Knapp and John A. Daly, "The SAGE Handbook of Interpersonal Communication", SAGE Publications, 2011, ISBN-10: 1412974747, ISBN-13: 978-1412974745.3.
8. John Stewart, "Bridges Not Walls: A Book About Interpersonal Communication", McGraw-Hill Education, 2011, ISBN-10: 0073534315, ISBN-13: 978-0073534312.
9. Pamela J. Kalbfleisch, "Interpersonal Communication: Evolving Interpersonal Relationships", Routledge, 2013, ISBN-10: 0805816611, ISBN-13: 978-0805816619.
10. Deborah Tannen, "Talking from 9 to 5: Women and Men at Work", William Morrow Paperbacks, 2001, ISBN-10: 0060959622, ISBN-13: 978-0060959623.

Mathematics-IIA (TIU-BS-UMA-T12101A)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 2nd Sem.
Course Title: Mathematics-II	**Subject Code:** TIU-BS-UMA-T12101A
Contact Hours/Week: 3–1–0 (L–T–P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. understand the basics of complex analysis.
2. understand algebraic and geometric representations of vectors and vector spaces and various operations on vector spaces.
3. solve differential equations with series solution method
4. learn the applications of the definite and indefinite integrals.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	analyze complex functions based on analyticity, integrability along a contour, calculus of residue, etc. and its applications in engineering.	K4
CO-2:	develop an understanding of vector spaces and inner product spaces.	K4
CO-3:	identify linear transformations on vector spaces and determine the corresponding matrix representation.	K4
CO-4:	determine the solution of ordinary differential equations using a series solution method.	K4
CO-5:	formulate some special functions, namely, Legendre and Bessel functions.	K4
CO-6:	develop an understanding of Integral calculus and its applications such as determining the area between two curves, the surface of revolution etc.	K4

COURSE CONTENT:

MODULE 1:	**Complex analysis**	**10 Hours**
Complex analysis: Limit, continuity, differentiability and analyticity of functions, Cauchy-Riemann equations, line integrals, Cauchy Goursat theorem (statement only), independence of path, Complex integration over a contour, Cauchy's integral formula, derivatives of analytic functions, Taylor's series, Laurent's series, Zeros and singularities, Residue theorem, evaluation of real integrals by contour integration.		
MODULE 2:	**Linear algebra**	**10 Hours**
Linear Algebra: Vector spaces over any arbitrary field, linear combination, linear dependence and independence, basis and dimension, linear transformations, matrix representation of linear transformations, linear functional, dual spaces, Inner product spaces, norms, Gram-Schmidt process, orthonormal bases, projections and least squares approximation.		
MODULE 3:	**Series solution of ODE**	**10 Hours**
Series solution of ODE: Review of power series, Ordinary point, regular and irregular singular point, series solution near ordinary and regular singular point. Legendre's equation and Legendre polynomials, Bessel's equation and Bessel's functions.		
MODULE 4:	**Integral calculus**	**8 Hours**
Riemann Integral, fundamental theorem of integral calculus, applications of definite integrals, improper integrals, Beta and Gamma functions, reduction formulae. Double and triple integration, change in order of integration, Jacobian and change of variables formula. Parametrization of curves and surfaces.		
MODULE 5:	**Vector calculus**	**7 Hours**
Vector fields, divergence and curl, Line integrals, Green's theorem, surface integral, Gauss and Stokes' theorems with applications.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO3	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO4	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO5	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO6	3	-	2	-	-	-	-	-	-	-	-	2	3	-	-
	3	2.2	2	2								2	3		

Books:

1. Higher Engineering Mathematics, *B. S. Grewal*
2. Advanced Engineering Mathematics, *Kreyszig*
3. A Text Book of Engineering Mathematics, *Rajesh Pandey*
4. Engineering Mathematics, *B. K. Pal, K. Das*

Mathematics for Data Science (TIU-BS-UMA-T12102)

Program: B.Tech CSE	**Year, Semester: 1**st Yr. 2nd Sem.
Course Title: Mathematics for Data Science	**Subject Code:** TIU-BS-UMA-T12102
Contact Hours/Week: 3-0-0 (L–T–P)	**Credit: 3**

COURSE OBJECTIVE:

Enable the student to:

1. understand algebraic and geometric representations of vectors and vector spaces and various operations on vector spaces and inner product space.
2. learn the basics of probability and apply them to real time problems.
3. understand basic statistics, dispersion, regression and curve fitting technique

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	develop an understanding of vector spaces and inner product spaces.	K4
CO-2:	identify linear transformations on vector spaces and to determine the corresponding matrix representation.	K4
CO-3:	calculate the probability using basic knowledge and fundamental concepts of probability.	K4
CO-4:	illustrate conditional probability, Bayes' Theorem and understand their scope of application to real world problems	K4
CO-5:	To investigate data-based on measures of central tendency, measures of dispersion	K4
CO-6:	To analyze observations in terms of regression and curve fitting	K4

COURSE CONTENT:

MODULE 1:	**Linear algebra**	**20 Hours**
Linear Algebra: Vector spaces over any arbitrary field, linear combination, linear dependence and independence, basis and dimension, linear transformations, matrix representation of linear transformations, linear functional, dual spaces, Inner product spaces, norms, Gram-Schmidt process, orthonormal bases, projections and least squares approximation.		

MODULE 2:	Basic Probability	10 Hours
Classical, relative frequency and axiomatic definitions of probability, mutually exclusive events, independent events, conditional probability, Bayes' Theorem.		
MODULE 3:	**Integral calculus**	**15Hours**
Raw data, Histogram, Frequency polygon. Measures of central tendencies – Arithmetic mean, Geometric mean, Harmonic mean, Weighted A.M., G.M. and H.M.; Mode, Median, Empirical relation between mean, median and mode; Mean, median and mode for grouped and ungrouped data. Measures of dispersion- standard deviation and variance for grouped and ungrouped data. Correlation and Regression – Covariance, Spearman's coefficient of correlation for grouped and ungrouped data; regression and least square curve fitting		
TOTAL HOURS:		**45 Hours**

Textbooks:

1. Higher Engineering Mathematics, B. S. Grewal
2. Advanced Engineering Mathematics, E. Kreyszig
3. Linear Algebra, S. H. Friedberg, A. J. Insel, L. E. Spence
4. Engineering Mathematics, B. K. Pal, K. Das

Engineering Mechanics (TIU-ES-UME-T12101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 2nd Sem.
Course Title: Engineering Mechanics	**Subject Code:** TIU-ES-UME-T12101
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. understand the basics of vector mechanics and its applications in engineering mechanics
2. analyze problems in statics
3. analyze problems in dynamics of particles

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	To understand the basics of vector mechanics and its application in engineering mechanics.	K2
CO-2:	To understand different force systems and the methods of finding their resultants and to be well-versed with the conditions of equilibrium in 2D.	K2

CO-3:	To be able to apply the laws of static equilibrium in solving problems and perform analysis of statically determinate trusses.	K4
CO-4:	To be able to compute centroids of plane areas, composite areas and to be able to compute area moments of inertias and radii of gyration of plane figures.	K3
CO-5:	To understand basic principles of kinematics of particles, plane, rectilinear and curvilinear coordinate systems and projectile motion	K3
CO-6:	To understand basic principles of kinetics of particles leading to Newton's laws and to be able to apply the work-energy and the linear impulse-linear momentum theorems in solving typical problems	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**4 Hours**
Introduction: Fundamentals of Mechanics: Introduction to mechanics; Basic concepts – mass, space, time and force; Particles and rigid bodies; Scalars and vectors; Free, sliding, fixed and unit vectors; Addition, subtraction and multiplication of two vectors; scalar triple product and vector product of 3 vectors.		
MODULE 2:	**FORCE SYSTEMS AND EQUILIBRIUM**	**9 Hours**
Force systems: Introduction to different force systems; Composition of forces – triangle, parallelogram and polygon law of forces, and addition of two parallel forces; Resolution of forces; Moment of a force, Varignon's theorem; Couples; Force-couple system; Resultant of a force system Equilibrium: Force Systems & Equilibrium: Free body diagram, equilibrium conditions in 2 dimensions, equilibrium of systems involving friction.		
MODULE 3:	**STRUCTURES**	**5 Hours**
Plane Truss: Statically determinate trusses; Force analysis of a truss - method of joints, method of sections		
MODULE 4:	**DISTRIBUTED FORCES**	**7 Hours**
Distributed Forces: Line, area and volume distributions of forces; Centre of gravity; Centre of mass; Centroids of plane figures; Centroids of composite areas. Moment of Inertia: Area moment of inertia; Perpendicular and Parallel axes theorems pertaining to moment of inertia; Radius of gyration.		
MODULE 5:	**KINEMATICS OF PARTICLES**	**8 Hours**
Kinematics of Particles: Differential equations of kinematics – plane, rectilinear and curvilinear motions; Cartesian co-ordinate system; Normal and tangent co-ordinate system, projectile motion.		
MODULE 6:	**KINETICS OF PARTICLES**	**12 Hours**
Kinetics of Particles: Newton's second law of motion; Work and energy principle – gravitational potential energy, elastic potential energy, kinetic energy, power, work-energy theorem, principle of impulse and momentum.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-

CO2	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	2	-	-	-	-	-	-	-	-	2	3	-	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	3	-	-
CO5	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO6	3	-	2	-	-	-	-	-	-	-	-	2	3	-	-
	3	2.4	2.3	2								2	3		

Books:

1. J. L. Meriam and L. G. Kraige, Engineering Mechanics (Vol.1 & 2), Wiley India 2017.
2. Shames I. H., Rao G. K. M., Engineering Mechanics, Pearson, 2005.
3. Khurmi R.S. ,A Textbook of Engineering Mechanics, S. Chand, 2018.
4. Bhavikatti S. S, Engineering Mechanics, New Age International Publishers, 2021.

Problem Solving Using Data Structures (TIU-ES-UCS-T12101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 2nd Sem
Course Title: Problem Solving Using Data Structures	**Subject Code:** TIU-ES-UCS-T12101
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVES:

1. Introduce fundamental data structures such as arrays, linked lists, stacks, queues, and trees, and their role in computational problem-solving.
2. Develop logical and analytical thinking by applying data structures to efficiently store, process, and manipulate data in various programming scenarios.
3. Enhance problem-solving abilities by selecting appropriate data structures based on efficiency, scalability, and real-world applicability.

COURSE OUTCOMES:

On completion of the course, the student will be able to:

CO-1	Recall and describe fundamental data structures, including arrays, linked lists, stacks, queues, and trees.	K1
CO-2	Explain searching and sorting techniques, along with their efficiency on different data structures.	K2
CO-3	Apply array and linked list operations to solve computational problems.	K3
CO-4	Implement stack and queue-based algorithms for expression evaluation and problem-solving scenarios.	K3
CO-5	Examine tree-based data structures (Binary Trees, BSTs) and their traversal techniques for problem-solving.	K4
CO-6	Compare different data structures based on their efficiency, scalability, and real-world applicability.	K4

COURSE CONTENT:

MODULE 1:	**BASIC CONCEPTS OF DATA REPRESENTATION**	**6 Hours**
Abstract Data Types, Fundamental and Derived Data Types, Representation, Primitive Data Structures.		
MODULE 2:	**ARRAYS**	**9 Hours**
Representation of Arrays, Single and Multidimensional Arrays, Address Calculation Using Column and Row Major Ordering, Various Operations on Arrays, Application of Arrays in Matrix Multiplication, Sparse Polynomial Representation and Addition. Solving different problems using Arrays: Find the missing number in an array, Rotate an array to the right by k steps by reversing the array and its sub-arrays, Move all zeros in the array to the end while maintaining the relative order of non-zero elements using a two-pointer approach.		
MODULE 3	**SEARCHING AND SORTING ON VARIOUS DATA STRUCTURES**	**6 Hours**
Sequential Search, Binary Search, Comparison-based sorting concepts, Bubble Sort, Insertion Sort, Selection Sort.		
MODULE 4	**STACKS AND QUEUES**	**9 Hours**
Representation of Stacks and Queues using Arrays and Linked List, Circular Queues. Applications of Stacks: Conversion from Infix to Postfix and Prefix Expressions, Evaluation of Postfix Expression Using Stacks. Solving different problems using stack and queue: Validates if parentheses are balanced, Finds the next greater element for each item in a stack, Implements stack operations using two queues, Reverses the elements of a queue, Implements queue operations using two stacks, Implements a circular queue, Implements queue operations using two stacks.		
Module 5	**Linked Lists**	**6 Hours**

Single Linked List, Operations on List, Polynomial Representation and Manipulation Using Linked Lists, Circular Linked Lists, Doubly Linked Lists. Solving different problems using Linked List: Reverse the order of elements in a singly linked list, Merge two linked lists into one list.		
Module 6	**Trees**	**9 Hours**
Binary Tree, Binary Search Tree, Traversal Methods: Preorder, In-Order, Post-Order Traversal (Recursive And Non-Recursive), Representation (Non-threaded and Threaded) of Trees and its Applications.		
	TOTAL LECTURE	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	2	-	-	-	-	-	-	-	3	-	-
CO2	3	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO3	2	3	2	-	-	-	-	-	-	-	-	-	-	-	3
CO4	2	3	2	-	-	-	-	-	-	-	-	-	-	-	3
CO5	2	2	3	-	-	-	-	-	-	-	-	-	-	-	3
CO6	2	3	-	-	2	-	-	-	-	-	-	-	-	3	-
	2.333	2.6	2.33		2								3	3	3

Books:

1. "Data Structures in C" by Tanenbaum, Moshe J. & Augenstein, PhilipC
2. Gilberg and Forouzan: “Data Structure- A Pseudo code approach with C” by Thomson publication
3. “Fundamental of Data Structure” (Schaums Series) Tata-McGraw-Hill.
4. “Fundamentals of data structure in C” Horowitz, Sahani & Freed, Computer Science Press.
5. "Data Structures Using C" by Reema Thareja

Basic Electrical & Electronics Engineering (TIU-ES-UEE-T12101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 2nd Sem.
Course Title: Basic Electrical & Electronics Engineering	**Subject Code:**TIU-ES-UEE-T12101
Contact Hours/Week: 3-1-0 (L-T-P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. Aanalyze and describe the basic electrical quantities, circuit elements, and their voltage-current relationships.

2. Design and analyze diode circuits, transistor biasing, and operational amplifier applications.
3. Understand the operation and characteristics of semiconductor devices like diodes, BJTs, JFETs, and MOSFETs.
4. Analyzing differential working principles of single-phase transformers, including voltage transformation and regulation.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Understand Basic Electrical Concepts	K2
CO-2:	Analyze DC Electrical Networks	K4
CO-3:	Analyze AC Circuits and Power Systems	K4
CO-4:	Understand Semiconductor Devices and Applications	K2
CO-5:	Design and Analyze Analog Circuits	K3
CO-6:	Understand Transformer Principles and Applications	K2

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**4 Hours**
Basic electrical quantities, Voltage, Current, Power. Basic Electrical elements: Resistance, Inductance,Capacitance. Their voltage-current relationship.Voltage and current sources.		
MODULE 2:	**DC NETWORK ANALYSIS**	**6 Hours**
KCL and KVL and their applications in purely resistive circuits.Concept of linear, bilateral networks.Sourceconversion, Star-Delta conversion.		
MODULE 3:	**DC NETWORK THEOREMS**	**5 Hours**
Superposition Theorem, Thevenin's Theorem, Norton's Theorem, Maximum Power Transfer Theorem.		
MODULE 4:	**SINUSOIDAL STEADY STATE ANALYSIS**	**5 Hours**
Matrix and Determinant: Revision of matrix and determinant, rank and nullity, solutions ofsystem of linear equations using Determinants and Matrices; Eigenvalues and eigenvectors, Cayley-Hamilton Theorem, transformation of matrices, adjoint of an operator, normal, unitary, hermitian and skew-hermitian operators, quadratic forms.		
MODULE 5:	**3-PH CIRCUITS**	**5 Hours**
Introduction to 3-Ph quantities.3-ph star and delta connection. Phasor diagram for 3-ph system, Balanced 3-phloads, measurement of 3-ph power.		
MODULE 6:	**SEMICONDUCTOR DEVICES**	**6 Hours**
Energy bands in solids.Intrinsic and extrinsic semiconductors.P-N junctions. Semiconductor diodes: ZenerandVaractor diodes. Bipolar transistors (operation, characteristics).		
MODULE 7:		**4 Hours**

Diode Circuits, BJT biasing & Operation of JFET, MOSFET		
MODULE 8:	**OPAMPS**	**5 Hours**
Properties of an ideal and a practical OPAMP. Block diagram. Concept of Virtual Short, Inverting and Non-inverting amplifiers, Summing and Differencing amplifier, Differentiator and Integrator.		
MODULE 9:	**1-Ph TRANSFORMERS**	**5 Hours**
Faraday's Law, EMF generation (dynamic and static), B-H curve, Construction and operation of single phasetransformer: voltage and current transformation, no-load operation, voltage regulation on resistive load.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	-	2	-	-	-	-	-	-	-	2	3	-	-
CO4	3	2	-	-	3	-	-	-	-	-	-	2	3	-	-
CO5	3	-	3	-	3	-	-	-	-	-	-	2	3	-	-
CO6	3	-	2	-	3	-	-	-	-	-	-	2	3	-	-
	3	2.5	2.5	2	3							2	3		

Books:

1. D. Chattopadhyay, P. C. Rakshit, Funndamentals of Electric Circuit Theory, S. Chand. Publications
2. D. Chattopadhyay, P.C. Rakshit, Electronics Fundamentals and Applications, New Age International Publisher
3. Salivahanan and P. Kumar**,** Circuit Theory, Vikas Publishing House
4. Kulshreshtha, Basic Electrical Engineering: Principles and Application, Tata McGraw-Hill.

Problem Solving Using Data Structures Lab (TIU-ES-UCS-L12101)

Program: B. Tech in CSE	**Year, Semester:** 1st Yr., 2nd Sem
Course Title: Problem Solving Using Data Structures Lab	**Subject Code:** TIU-ES-UCS-L12101
Contact Hours/Week: 0–0–3 (L–T–P)	**Credit:** Lab.–1.5

Course Objective:

Enable the students to

1. Develop a strong foundation in data structures and algorithms with a focus on both linear and non-linear structures.
2. Implement and analyze searching, sorting, and graph algorithms to optimize problem-solving efficiency.
3. Enhance programming skills by applying data structures in real-world applications and evaluating their complexity.
4. Understand and assess the time and space complexity of algorithms for efficient software development.

Course Outcome:

CO-1	Understand fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs along with their applications.	K2
CO-2	Implement various data structures using programming techniques to efficiently store, manipulate, and retrieve data.	K3
CO-3	Analyze and apply different searching and sorting algorithms to optimize problem-solving.	K4
CO-4	Evaluate the time and space complexity of algorithms to improve computational efficiency.	K5
CO-5	Apply data structures and algorithms to solve real-world problems and develop efficient software solutions.	K3
CO-6	Explore advanced data structures and algorithmic techniques for tackling complex computing challenges.	K6

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**6 Hours**
Basic Concepts of Data Representation: Abstract Data Types, Fundamental and Derived Data Types, Representation, Primitive Data Structures.		
MODULE 2:	**ARRAY REPRESENTATION**	**6 Hours**
Arrays: Representation of Arrays, Single and Multidimensional Arrays, Address Calculation Using Column and Row Major Ordering, Various Operations on Arrays, Application of Arrays Matrix Multiplication, Sparse Polynomial Representation and Addition. Solving different problems using Arrays such as the followings: Find the missing number in an array, Rotate an array to the right by k steps by reversing the array and its sub-arrays, Move all zeros in the array to the end while maintaining the relative order of non-zero elements using a two-pointer approach.		
MODULE 3:	**SEARCHING AND SORTING TECHNIQUES**	**6 Hours**
Searching and Sorting on Various Data Structures: Sequential Search, Binary Search, Comparison based sorting concept, Bubble sort, Insertion Sort, Selection Sort.		
MODULE 4:	**STACK AND QUEUE**	**9 Hours**
Stacks and Queues: Representation of Stacks and Queues using Arrays and Linked List, Circular Queues. Applications of Stacks, Conversion from Infix to Postfix and Prefix Expressions, Evaluation of Postfix Expression Using Stacks. Solving different problems using stack and queue such as		

Validates if parentheses are balanced, Finds the next greater element for each item in a stack, Implements stack operations using two queues, Reverses the elements of a queue, Implements queue operations using two stacks, Implements a circular queue, Implements queue operations using two stacks.		
MODULE 5:	**LINKED LISTS**	**9 Hours**
Linked Lists: Single Linked List, Operations on List, Polynomial Representation and Manipulation Using Linked Lists, Circular Linked Lists, Doubly Linked Lists. Solving different problems using Linked List such as Reverse the order of elements in a singly linked list, Merge two linked lists into one list.		
MODULE 6:	**TREE DATA STRUCTURES AND TRAVERSALS**	**9 Hours**
Trees: Binary Tree, Binary Search Tree, Traversal Methods: Preorder, In-Order, Post-Order Traversal (Recursive And Non-Recursive), Representation (Non-threaded and Threaded) of Trees and its Applications.		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	2	-	-	-	-	-	-	-	3	-	-
CO2	2	-	3	-	3	-	-	-	-	-	-	-	-	-	3
CO3	3	3	-	-	2	-	-	-	-	-	-	-	-	-	3
CO4	3	2	-	-	2	-	-	-	-	-	-	-	-	3	-
CO5	2	3	3	-	-	-	-	-	-	-	2	-	-	2	3
CO6	2	3	-	2	2	-	-	-	-	-	-	2	-	3	3
	2.5	2.75	3	2	2.2						2	2	3	2.666	3

Books:

1. "Data Structures in C" by Tanenbaum, Moshe J. & Augenstein, PhilipC
2. Gilberg and Forouzan: “Data Structure- A Pseudocode approach with C” by Thomson publication
3. “Fundamentals of Data Structure” (Schaum's Series) Tata-McGraw-Hill.
4. “Fundamentals of data structure in C” Horowitz, Sahani & Freed, Computer Science Press.
5. "Data Structures Using C" by Reema Thareja

Basic Electrical Engineering Lab and Simulation (TIU-ES-UEE-L12101)

Program: B. Tech. in CSE	**Year, Semester:** 1st Yr., 2nd Sem.
Course Title: Basic Electrical Engineering Lab and Simulation	**Subject Code:** TIU-ES-UEE-L12101

Contact Hours/Week: 0–0–3 (L–T–P)	**Credit: 1.5**

COURSE OBJECTIVE:

Enable the student to:

1. introduce fundamental electrical and electronic circuit theorems and develop analytical skills for solving electrical networks.
2. familiarize students with essential circuit components, including R-L-C circuits, diodes, rectifiers, and fluorescent lamps, and their practical applications.
3. enhance hands-on laboratory skills by conducting experiments on circuit analysis, diode characteristics, and rectifier efficiency evaluation.

COURSE OUTCOME:

The students will be able to:

CO-1	Identify and understand fundamental electrical and electronic circuit theorems and their applications.	K1
CO-2	Explain the working principles of R-L-C circuits, diodes, rectifiers, and fluorescent lamps.	K2
CO-3	Apply circuit theorems such as Superposition and Thevenin's Theorem to analyze electrical networks.	K3
CO-4	Conduct experiments to measure and analyze V-I characteristics of P-N junction and Zener diodes.	K3
CO-5	Evaluate the efficiency and power factor of electrical circuits, rectifiers, and fluorescent lamps.	K4
CO-6	Compare different rectifier circuits and analyze their output waveforms and ripple factors.	K4

COURSE CONTENT:

Experiment 1	**Verification of Superposition Theorem**	**3 Hours**
Theoretical foundation of superposition theorem, Application in linear electrical circuits, Step-by-step circuit analysis with multiple voltage/current sources, Practical applications in circuit design, troubleshooting, and network analysis.		
Experiment 2	**Study of R-L-C Series Circuit**	**3 Hours**
Characteristics of resistance (R), inductance (L), and capacitance (C) in AC circuits, Impedance (Z) and phase angle, Voltage and current phase relationships, Leading and lagging power factor, Practical applications in circuit analysis and troubleshooting.		
Experiment 3	**Verification of Thevenin's Theorem**	**3 Hours**
Theoretical foundation of Thevenin's theorem, Converting complex circuits into Thevenin equivalent, Measuring Thevenin voltage (Vth) and resistance (Rth), Practical applications in circuit design and network analysis.		
Experiment 4	**Characteristics of Fluorescent Lamp**	**3 Hours**
Gas discharge and phosphor coating in light production, Role of starter, choke (ballast), and		

electrodes, Measuring voltage, current, and power consumption, Efficiency comparison with incandescent and LED lamps, Impact of inductive ballast on power factor and improvement methods, Performance comparison of electromagnetic vs. electronic ballasts, Energy savings, lifespan, and environmental concerns (mercury content).		
Experiment 5	**Familiarization with Basic Electronic Components**	**3 Hours**
Identification, specifications, and testing of R, L, and C components (Color codes), Potentiometers, switches (SPDT, DPDT, DIP), Breadboards and Printed Circuit Boards (PCBs), Active components: Diodes, BJTs, JFETs, MOSFETs, Power transistors, SCRs, LEDs.		
Experiment 6	**Study of V-I Characteristics of P-N Junction Diode in Forward Bias**	**3 Hours**
Depletion layer and barrier potential, Forward bias operation, Breakdown voltage and Peak Inverse Voltage (PIV), Knee voltage and ideal PN junction diode characteristics.		
Experiment 7	**V-I Characteristics of Zener Diode in Reverse Bias**	**3 Hours**
Depletion layer and barrier potential, Reverse bias operation, Breakdown voltage and Peak Inverse Voltage (PIV), Knee voltage and ideal Zener diode characteristics.		
Experiment 8	**Study of Half-Wave and Full-Wave Rectifier**	**3 Hours**
Half-wave and full-wave rectifiers (Center-tap and Bridge), Output waveforms and voltage regulation, Ripple factor and rectifier efficiency.		
	TOTAL LAB HOURS	**24 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	-	2	-	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	-	2	-	-	-	-	-	-	-	-	3	-	-
CO4	3	-	-	3	-	-	-	-	-	-	-	2	3	-	-
CO5	3	-	-	2	3	-	-	-	-	-	-	2	3	-	-
CO6	3	-	-	2	3	-	-	-	-	-	-	2	3	-	-
	3	2.5	2	2.3	3							2	3		

Books:

1. Boylestad, R. L., & Nashelsky, L. (2015). Electronic devices and circuit theory (11th ed.). Pearson.

2. Hayt, W. H., Kemmerly, J. E., & Durbin, S. M. (2018). Engineering circuit analysis (9th ed.). McGraw-Hill Education.
3. Sedra, A. S., & Smith, K. C. (2016). Microelectronic circuits (7th ed.). Oxford University Press.
4. Malvino, A. P., & Bates, D. J. (2016). Electronic principles (8th ed.). McGraw-Hill Education.

SEMESTER 3

MATHEMATICS-III (DISCRETE MATHEMATICS)
(TIU-UMA-T215)

Program: B. Tech. in CSE	**Year, Semester:** 2nd Yr., 3rd Sem.
Course Title: MATHEMATICS-III (DISCRETE MATHEMATICS)	**Subject Code:** TIU-UMA-T215
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. develop a foundation of set theory and concept of mathematical induction.
2. explore groups, rings and fields.
3. develop logical reasoning techniques and notation, demonstrate the application of logic to analyzing and writing proofs.
4. develop techniques for counting, permutations and combinations.
5. explore the concept of recurrence relations and generating functions and applications in algorithms.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Develop a foundation of set theory and concept of mathematical induction	K3
CO-2:	Analyse the concepts of a group with examples.	K4
CO-3:	Develop formal logical reasoning techniques and notation, demonstrate the application of logic to analyze and write proofs	K4
CO-4:	Construct techniques for counting, permutations and combinations.	K4
CO-5:	Apply the concept of recurrence relations and generating functions and applications in algorithms	K4
CO-6:	Examine if algebraic structures are rings or fields.	K4

COURSE CONTENT:

MODULE 1:	**Propositional logic**	**10 Hours**
Logical operators, propositional equivalences, normal forms, validity and satisfiability of arguments. Proof techniques: forward proof, proof by contradiction, contrapositive proofs, proof of necessity and sufficiency.		
MODULE 2:	**Sets, relations and functions**	**6 Hours**
Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction. Size of a set: Finite and infinite sets, countable and uncountable sets.		
MODULE 3:	**Introduction to counting**	**10 Hours**
Basic counting techniques - inclusion and exclusion, pigeon-hole principle, permutation,		

combination, summations.		
MODULE 4:	**Recurrence**	**9 Hours**
Introduction to recurrence relation and generating function, Tower of Hanoi, Fibonacci Series. Derangement – Hatcheck Problem.		
MODULE 5:	**Algebraic structures and morphisms**	**10 Hours**
Algebraic structures with one binary operation - semigroups, monoids and groups, congruence relation and quotient structures. Free and cyclic monoids and groups, permutation groups, substructures, normal subgroups. Algebraic structures with two binary operations - rings, integral domains and fields.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	-	2	-	-	-	-	-	-	-	2	3	2	2
CO4	3	3	-	-	-	-	-	-	-	-	-	2	3	-	-
CO5	3	3	-	2	-	-	-	-	-	-	-	3	3	3	3
CO6	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
	3	2.5		2								2.167	3	2.5	2.5

Books:

1. Discrete Mathematics and Its Applications, K.H. Rosen.
2. Discrete Mathematics: An Open Introduction, O. Levin.

Environmental Science (TIU-UMB-T201)

Program: B. Tech in CSE	**Year, Semester:** 2nd Yr., 3rd Sem
Course Title: Environmental Science	**Subject Code:** TIU-UMB-T201
Contact Hours/Week: 2–0–0(L–T–P)	**Credit:** 2

COURSE OBJECTIVE:

Enable the student to:

1. Understand the fundamentals of environmental pollution from chemical processes, including characterization of emissions and effluents, and relevant environmental regulations.
2. Apply pollution prevention strategies through process modification, resource recovery, and waste minimization techniques.

3. Analyze and design air and water pollution control systems, including particulate and gaseous emission control, and physical water treatment processes.
4. Evaluate and implement biological treatment methods for wastewater and appropriate solid waste disposal techniques.

COURSE OUTCOME:

The students will be able to:

CO-1:	Sources & types of pollution, industrial emissions & effluents, environmental laws & standards	K2
CO-2:	Pollution prevention, waste recovery & reuse, material & energy balance, water & emission control	K3
CO-3:	Selection & design of particulate and gaseous emission control systems, equipment performance analysis	K4
CO-4:	Wastewater treatment principles, solids removal processes (sedimentation, filtration, coagulation, etc.)	K4
CO-5:	Biological treatment principles, biochemical kinetics, aeration & sludge separation design	K4
CO-6:	Solid waste disposal methods, briquetting & gasification	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**6 Hours**
Environment and environmental pollution from chemical process industries, characterization of emission and effluents, environmental Laws and rules, standards for ambient air, noise emission and effluents.		
MODULE 2:	**POLLUTION PREVENTION**	**5 Hours**
Process modification, alternative raw material, recovery of by co-product from industrial emission effluents, recycle and reuse of waste, energy recovery and waste utilization. Material and energy balance for pollution minimization. Water use minimization, Fugitive emission/effluents and leakages and their control-housekeeping and maintenance.		
MODULE 3:	**AIR POLLUTION CONTROL**	**7 Hours**
Particulate emission control by mechanical separation and electrostatic precipitation, wet gas scrubbing, gaseous emission control by absorption and adsorption, Design of cyclones, ESP, fabric filters and absorbers		
MODULE 4:	**WATER POLLUTION CONTROL**	**6 Hours**

Physical treatment, pre-treatment, solids removal by setting and sedimentation, filtration centrifugation, coagulation and flocculation.		
MODULE 5:	**BIOLOGICAL TREATMENT**	**3 Hours**
Anaerobic and aerobic treatment biochemical kinetics, trickling filter, activated sludge and lagoons, aeration systems, sludge separation and drying.		
MODULE 6:	**SOLID DISPOSAL**	**3 Hours**
Solids waste disposal - composting, landfill, briquetting / gasification and incineration.		
	TOTAL LECTURES	**30 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	3	-	-	-	-	2	2	-	-
CO2	3	2	-	-	-	-	3	-	-	-	-	2	2	-	-
CO3	3	2	-	-	-	-	2	-	-	-	-	2	2	-	-
CO4	3	3	-	-	-	-	2	-	-	-	-	-	2	-	-
CO5	3	3	-	-	-	-	-	-	-	-	-	2	2	-	-
CO6	3	2	-	-	-	-	3	-	-	-	-	-	2	-	-
	3	2.33					2.6					2	2		

Books:

1. A. K. De, “Environmental Chemistry”, New Age
2. G. M. Masters, “Introduction to Environmental Engineering and Science”, Pearson
3. 3. G. S. Sodhi, “Fundamental Concepts of Environmental Chemistry”, Narosa
4. E. Odum, M. Barrick& G. W. Barrett, “Fundamentals of Ecology”, Brooks

Digital Electronics (TIU-UEC-T211)

Program: B. Tech. in CSE	**Year, Semester:** 2nd Yr., 3rd Sem.
Course Title: Digital Electronics	**Subject Code:** TIU-UEC-T211
Contact Hours/Week: 3–1–0 (L–T–P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. Understand number systems, arithmetic operations, conversions, binary codes, Boolean

algebra, logic gates, minimization techniques, and logic families.
2. Analyze combinational circuits (adders, multiplexers, encoders, etc.) and sequential circuits (flip-flops, registers, counters).
3. Explore memory devices (ROM, RAM, FPGA, etc.) and finite state machines, including Moore & Mealy models and state minimization.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Understand Boolean algebra, logic gates, number systems, codes, and logic families in digital systems.	K1
CO-2:	Apply minimization techniques to optimize digital circuits for real-time applications.	K3
CO-3:	Analyze and design combinational circuits for digital system implementation.	K2
CO-4:	Design modular combinational circuits using MUX/DEMUX, decoders, and encoders	K2
CO-5:	Understand and implement synchronous sequential circuits in digital systems.	K2
CO-6:	Explore memory elements, programmable devices, and use HDL & EDA tools for digital design and simulation.	K4

COURSE CONTENT:

MODULE 1:	**NUMBER SYSTEM AND CODES**	**6 Hours**
Number System and Codes: Decimal, binary, octal and hexadecimal number systems and their arithmetic operations, conversion of one number system to another, Signed and floating point representations of binary numbers, 1's complement and 2's complement representations, Binary codes, natural BCD codes, Excess-3, Gray codes, Alphanumeric codes, code conversion- from one code to another.		
MODULE 2:	**LOGIC GATES, BOOLEAN ALGEBRA & BASIC LOGIC FAMILIES**	**12 Hours**
Logic Gates, Boolean Algebra & Basic logic families: NOT, AND, OR, NAND, NOR, XOR and XNOR – operations, truth tables and Venn diagram representations, universal gates, postulates and laws of Boolean algebra, De Morgan's theorem, minterms and maxterms, SOP and POS forms, Switching algebra, Minimizing functions using Kmaps, Minimization using QM method, Different logic families: TTL, ECL.		
MODULE 3:	**COMBINATIONAL AND ARITHMETIC LOGIC CIRCUITS**	**7 Hours**
Combinational and arithmetic logic circuits: Adders/subtractors circuit using logic gates, fast adder, magnitude comparator, multiplexer demultiplexers, encoders, decoders, priority encoders, parity generator and checkers, BCD adder and subtractor.		
MODULE 4:	**SEQUENTIAL LOGIC CIRCUITS**	**8 Hours**
Sequential Logic Circuits: Flip flops and latches, S-R, J-K, D and T type flip-flops and their conversions, master-slave configuration, edge triggered and level triggered clock, registers, shift registers, synchronous and asynchronous counters, ring and Johnson (twisted ring) counters, Modulus Counters.		
MODULE 5:	**MEMORY AND PROGRAMMABLE LOGIC DEVICES**	**6 Hours**
Memory and Programmable Logic Devices: ROM, PROM, RAM-SRAM, DRAM, EPROM, EEPROM, Flash ROM, Programmable and gated array devices for designing combinational circuits PAL,		

PLA, PLD, CPLD, FPGA with examples.		
MODULE 6:	**FINITE STATE MACHINES**	**6 Hours**
Finite State Machines: Finite state machine state transition diagrams and state transition tables, Moore & Mealy machine state diagram, state variable, state table and state minimization, design of state machines using combinational logic circuits and memories.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATTIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	2	-	-	-	-	-	-	-	1	-	-
CO2	2	3	-	-	2	-	-	-	-	-	-	-	-	-	-
CO3	2	3	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	2	-	3	-	1	-	-	-	-	-	-	-	1	-	-
CO5	2	-	3	-	-	-	-	-	-	-	-	-	-	-	2
CO6	-	-	-	-	3	-	-	-	-	-	-	2	2	-	3
	2.2	3	3		1.8							2	1.333		2.5

Books:

1. D. P. Leach and A. Malvino, "Digital Principles and Applications", 7th Edi. McGraw Hill.
2. M. Morris Mano & M. D. Ciletti, "Digital Design", 3rd Edition, Prentice Hall.
3. A. Anand Kumar, "Fundamentals of Digital Circuits", Prentice Hall.
4. S. Salivahanan and S. Arivazhagan, "Digital Circuits & Design", Vikas.
5. D. L. Schilling and H. Taub, "Digital Integrated Electronics", McGraw Hill.
6. J. Bhaskar, "A VHDL Primer", Pearson.

Data Structure and Algorithms (TIU-UCS-T201)

Program: B. Tech. in CSE	**Year, Semester:** 2nd Yr., 3rd Sem
Course Title: Data Structure and Algorithms	**Subject Code:** TIU-UCS-T201
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

1. Introduce fundamental data structures, including linear and nonlinear structures, and their real-world applications to help students understand their significance in problem-solving.
2. Enable students to develop and implement various data structures and algorithms efficiently using programming techniques.
3. Equip students with the ability to analyze the time and space complexity of algorithms and

make informed trade-offs for optimizing performance.

4. Encourage students to apply data structures and algorithmic principles to design and develop efficient solutions for real-world computational problems.

COURSE OUTCOME:

The students will be able to:

CO-1	Understand the concepts and applications of different types of data structures	K2
CO-2	Develop programs to implement linear and nonlinear data structures	K3
CO-3	Be able to learn various algorithms and their implementations	K3
CO-4	Analyze algorithms to do efficiency tradeoffs	K4
CO-5	Apply the concepts of data structures and algorithms to find efficient solutions for real world problems	K3
CO-6	Understand the concepts of complex data structures and algorithms.	K5

COURSE CONTENT:

MODULE 1:	**Fundamentals of DSA**	**6 Hours**
Basic Concepts of Data Representation: Abstract Data Types, Fundamental and Derived Data Types, Representation, Primitive Data Structures. Introduction to Algorithm Design and Data Structures: Algorithm Definition, Comparison of Algorithms, Top-Down and Bottom-Up Approaches to Algorithm Design, Analysis of Algorithm, Complexity Measures in Terms of Time and Space, Structured Approach to Programming.		
MODULE 2:	**Arrays**	**10 Hours**
Arrays: Representation of Arrays, Single and Multidimensional Arrays, Address Calculation Using Column and Row Major Ordering, Various Operations on Arrays, Application of Arrays Matrix Multiplication, Sparse Polynomial Representation and Addition.		
MODULE 3:	**Stack, Queue and LinkedList**	**11 Hours**
Stacks and Queues: Representation of Stacks and Queues using Arrays and Linked List, Circular Queues, Priority Queue and D-Queue. Applications of Stacks, Conversion from Infix to Postfix and Prefix Expressions, Evaluation of Postfix Expression Using Stacks. Linked Lists: Single Linked List, Operations on List, Linked Stacks and Queues, Polynomial Representation and Manipulation Using Linked Lists, Circular Linked Lists, Doubly Linked Lists.		
MODULE 4:	**Trees and Graphs**	**10 Hours**
Trees: Binary Tree, Traversal Methods: Preorder, In-Order, Post-Order Traversal (Recursive and Non-Recursive), Algorithms for Above Mentioned Traversal Methods. Representation of Trees and Its Applications. Binary Tree. Binary Search Tree, Height Balanced (AVL) Tree, B-Trees, B+ Tree, Min Heap, Max Heap Graphs: Graph Representation, Adjacency Matrix, Adjacency Lists, Traversal Schemes, Depth First Search, Breadth First Search.		

MODULE 5:	**Searching, Sorting and Hashing**	**8 Hours**
Searching, Sorting and Complexity: Searching: Sequential and Binary Searches, Indexed Search, Hashing Schemes. Sorting: Insertion, Selection, Bubble, Quick, Merge.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	1	-	-	-	-	-	-	-	2	-	-
CO2	2	-	3	-	2	-	-	-	-	-	-	-	2	-	2
CO3	2	2	-	-	2	-	-	-	-	-	-	-	-	-	-
CO4	1	3	-	2	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	3	-	2	-	-	-	-	-	-	-	-	2	3
CO6	2	-	-	-	3	-	-	-	-	-	-	2	1	2	3
	2	2.5	3	2	2							2	1.66	2	2.66

Books:

1. "Data Structures in C" by Tanenbaum, Moshe J. & Augenstein, PhilipC
2. Gilberg and Forouzan: "Data Structure- A Pseudo code approach with C" by Thomson publication
3. "Fundamental of Data Structure" (Schaums Series) Tata-McGraw-Hill.
4. "Fundamentals of data structure in C" Horowitz, Sahani & Freed, Computer Science Press.
5. "Data Structures Using C" by Reema Thareja

Computer Organization (TIU-UCS-T207)

Program: B. Tech. in CSE	**Year, Semester:** 2nd Yr., 3rd Sem
Course Title: Computer Organization	**Subject Code:** TIU-UCS-T207
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Understand fundamental blocks of computer architecture
2. Understand memory Hierarchy and how does it help in system performance
3. Understand pipelining techniques to improve system performance
4. Understand CPU and CU design

COURSE OUTCOME:

The students will be able to:

CO 1:	Describe Stored Program Digital Computer System	K1
CO 2:	Identify & apply appropriate procedures and algorithms of Computer Arithmetic	K3
CO 3:	Explain different aspects of Central Processing Unit (CPU).	K3
CO 4:	Understand the fundamentals of Memory Unit and illustrate memory operations.	K2
CO 5:	Explain models of I/O operations & the I/O subsystems.	K3
CO 6:	Identify the micro-instructions and basics of Computer Architecture.	K4

COURSE CONTENT:

MODULE 1:	**BASIC FUNCTIONAL BLOCKS OF A COMPUTER**	**10 Hours**
Von Neumann machines, Harvard Architecture, SISD, MISD, MIMD, Single instruction multiple data stream (SIMD) architectures concept of operating systems and processes, processor register sets, processor instruction sets, processor architecture, memory hierarchy, Parallel Processor and Pipeline Architecture.		
MODULE 2:	**DATA REPRESENTATION & COMPUTER ARITHMATIC**	**7 Hours**
Data Representation: Signed number representation, fixed and floating-point representations, Computer arithmetic - integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication -Booth multiplier, Division - non-restoring and restoring techniques.		
MODULE 3:	**CPU AND CONTROL UNIT DESIGN**	**7 Hours**
CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU - registers, instruction execution cycle, Basic Instruction format, Immediate, Direct address, Indirect address, Effective addresses. Instruction Formats: Memory/ Register/Input-Output reference, Types of Instruction: Data Transfer/ Data Manipulation/ Program Control, Zero/One/Two/Three address instructions, RISC instructions, RTL interpretation of instructions, addressing modes, instruction set. Case study - instruction sets of some common CPUs.		
MODULE 4:	**MEMORY ORGANISATION**	**7 Hours**
Concept of hierarchical memory organization, Memory interleaving, Semiconductor memory technologies, primary memory and concept of cache memory.		
MODULE 5:	**PERIPHERAL DEVICES AND THEIR CHARACTERISTICS**	**7 Hours**
Peripheral Devices and Their Characteristics: Input-output subsystems, I/O transfers – program controlled, interrupt driven and DMA		

MODULE 6:	**PIPELINEING**	**7 Hours**
Basic concepts of pipelining, throughput and speedup, pipeline hazards. (Parallel Processor)		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	1	-	-	-	-	-	-	-	2	-	-
CO2	2	2	-	-	2	-	-	-	-	-	-	-	1	-	-
CO3	2	-	2	-	1	-	-	-	-	-	-	-	1	-	-
CO4	2	-	2	-	-	-	-	-	-	-	-	-	1	-	-
CO5	2	-	-	-	3	-	-	-	-	-	-	2	2	-	-
CO6	3	-	2	-	-	-	-	-	-	-	-	-	2	-	-
	2.33	2	2		1.8							2	1.5		

Books:

1. Computer System Architecture M. M. Mano:, 3rd ed., Prentice Hall of India, New Delhi, 1993.
2. Computer Organization and Design: The Hardware/Software Interface, David A. Patterson and John L. Hennessy.
3. Computer Organization and Embedded Systems, Carl Hamacher.
4. Computer Architecture and Organization, John P. Hayes.
5. Computer Organization and Architecture: Designing for Performance, William Stallings.
6. Computer System Design and Architecture, Vincent P. Heuring and Harry F. Jordan

Computer organization Lab (TIU-UCS-L207)

Program: B.Tech. in CSE	**Year Semester:** 2nd, 3rd. Sem
Course Title: Computer organization Lab	**Subject Code:** TIU-UCS-L207
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand the fundamental principles of computer organization and digital system design.
2. Implement logic circuits and understand their role in computing systems.
3. Design and analyze basic components of computer architecture, including ALU and memory.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Describe the fundamental aspects of computer organization design and analysis.	K3
CO-2	Explain the basic hardware components using Integrated Circuits in this lab course.	K3
CO-3	Identify the different logic gates and demonstrate their applications.	K3
CO-4	Implement and apply hardware realizations across various domains	K4
CO-5	Design the memory unit of a computer system.	K4
CO-6	Construct an Arithmetic Logic Unit (ALU) for a computer system	K3

COURSE CONTENT:

MODULE 1:	**BASICS OF DIGITAL LOGIC**	**9 Hours**
Introduction to number systems, Boolean algebra, Logic gates and truth tables, Simplification using K-map		
MODULE 2:	**COMBINATIONAL CIRCUITS**	**9 Hours**
Design and implementation of Half adder, Full adder, Multiplexers, and Decoders		
MODULE 3:	**SEQUENTIAL CIRCUITS**	**6 Hours**
Introduction to flip-flops (SR, D, JK, T), Registers, Counters, and their applications		
MODULE 4:	**MEMORY ORGANIZATION**	**6 Hours**
RAM and ROM structures, Cache memory, Design of basic memory unit		
MODULE 5:	**ARITHMETIC LOGIC UNIT (ALU) DESIGN**	**6 Hours**
Design of a simple ALU, Implementation of arithmetic operations (addition, subtraction, multiplication)		
MODULE 6:	**CPU ORGANIZATION AND PIPELINING**	**9 Hours**
Basic CPU architecture, Instruction execution cycle, Introduction to pipelining		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	1	-	-	-	-	-	-	-	2	-	-
CO2	2	-	-	-	3	-	-	-	-	-	-	2	2	-	-

CO3	2	-	-	-	2	-	-	-	-	-	-	-	1	-	-
CO4	2	2	3	-	-	-	-	-	1	-	-	-	2	-	-
CO5	2	-	3	-	-	-	-	-	-	-	-	-	2	-	-
CO6	3	-	2	-	-	-	-	-	-	-	-	-	2	-	-
	2.33	2	2.67		2				1			2	1.833		

Books:

1. M. Morris Mano & Michael D. Ciletti - Digital Design (5th ed.), Pearson

Digital Electronics Lab (TIU-UEC-L211)

Program: B.Tech. in CSE	**Year, Semester:** 2nd Yr, 3rd Sem
Course Title: Digital Electronics Lab	**Subject Code:** TIU-UEC-L211
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Study basic and universal logic gates and verify their truth tables through practical implementation.
2. Design arithmetic circuits, multiplexers, decoders, and code converters.
3. Implement flip-flops and counters to understand memory storage and sequential logic.
4. Gain hands-on experience in troubleshooting and designing digital electronic circuits.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Understand the fundamental concepts of logic gates and their operations.	K2
CO-2	Implement and Analyze logic circuits using universal gates and verify their truth tables.	K4
CO-3	Analyze and Design combinational circuits and validate their functionalities.	K4
CO-4	Apply the working principles of basic sequential circuits.	K3
CO-5	Design and Develop optimized digital circuits for real-world applications using logic gates and sequential elements.	K6
CO-6	Evaluate the performance of various digital circuits and troubleshoot faults in hardware implementation.	K5

COURSE CONTENT:

MODULE 1:	**BASIC LOGIC GATES**	**9 Hours**
Basic gates (AND, OR, NOT), universal gates (NAND, NOR), and exclusive gates (XOR, XNOR)		

MODULE 2:	**DATA REPRESENTATION & CONVERSION**	**6 Hours**
Even and odd parity generation and checking, Binary to Gray and Gray to Binary Conversion		
MODULE 3:	**ARITHMETIC CIRCUITS**	**9 Hours**
Half Adder, Full Adder, Half Subtractor, and Full Subtractor circuits		
MODULE 4:	**COMBINATIONAL LOGIC CIRCUITS**	**9 Hours**
Multiplexer, decoder		
MODULE 5:	**SEQUENTIAL CIRCUITS - FLIP-FLOPS**	**6 Hours**
Different types of flip-flops (SR, JK, D, and T)		
MODULE 6:	**SEQUENTIAL CIRCUITS - COUNTERS**	**6 Hours**
Asynchronous and synchronous counters		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	1	-	-	-	-	-	-	-	2	-	-
CO2	2	-	-	-	3	-	-	-	-	-	-	2	2	-	-
CO3	2	-	3	-	2	-	-	-	-	-	-	-	2	-	-
CO4	2	-	2	-	1	-	-	-	-	-	-	-	-	-	-
CO5	2	-	3	-	-	-	-	-	1	-	-	-	2	-	-
CO6	2	2	-	2	-	-	-	-	-	-	-	-	-	-	-
	2.17	2	2.67	2	1.8				1			2	2		

Books:

1. M. Morris Mano & Michael D. Ciletti – Digital Design (5th Edition) PEARSON.
2. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss – Digital Systems: Principles and Applications (12th Edition) PEARSON.
3. A. Anand Kumar – Fundamentals of Digital Circuits (4th Edition)PHI.
4. S Salivahanan, S Arivazhagan - Digital Circuits And Design (5th Edition) OXFORD UNIVERSITY PRESS

Data Structures and Algorithms Lab (TIU-UCS-L271)

Program: B.Tech. in CSE	**Year, Semester:** 2nd, 3rd.
Course Title: Data Structures and Algorithms Lab	**Subject Code:** TIU-UCS-L209
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Develop and implement fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs to efficiently store and manage data.
2. Analyze and apply searching and sorting algorithms, including linear search, binary search, and various sorting techniques, to optimize computational performance.
3. Enhance problem-solving skills by utilizing recursion and advanced data structures in real-world applications, ensuring efficient algorithmic design and implementation.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Describe the fundamental concepts of data structures.	K3
CO-2	Implement structures and pointers in programming tasks.	K3
CO-3	Differentiate between various types of data structures (linear and nonlinear) and apply them in practical scenarios.	K4
CO-4	Explain the concept and workings of recursion.	K3
CO-5	Compare linear and binary search algorithms.	K4
CO-6	Master various sorting algorithms and apply them in problem-solving tasks.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO LINKED LISTS**	**6 Hours**
Concept of dynamic memory allocation and pointers; Singly Linked List: Implementation of creation, insertion, deletion, and traversal operations; Doubly Linked List: Implementation of creation, insertion, deletion, and traversal operations; Circular Linked List: Implementation of creation, insertion, deletion, and traversal operations.		
MODULE 2:	**STACK AND QUEUE IMPLEMENTATION**	**6 Hours**
Stack: Concept, applications (Expression evaluation, Backtracking), Implementation of stack operations using arrays and pointers; Queue: Concept, types (Linear Queue, Circular Queue, Priority Queue), Implementation of queue operations using arrays and pointers.		
MODULE 3:	**SEARCHING TECHNIQUES**	**6 Hours**
Linear Search: Implementation using iterative and recursive functions; Binary Search: Implementation using iterative and recursive functions; Performance analysis of searching algorithms.		
MODULE 4:	**SORTING TECHNIQUES**	**9 Hours**
Bubble Sort: Concept and implementation; Selection Sort: Concept and implementation; Insertion Sort: Concept and implementation; Comparative analysis of sorting algorithms based on time complexity.		
MODULE 5:	**TREE DATA STRUCTURES AND TRAVERSALS**	**9 Hours**
Introduction to Trees: Binary Trees, Binary Search Trees (BST); Tree Traversal Methods: Inorder, Preorder, Post order traversal implementation; Recursive and iterative approaches.		
MODULE 6:	**GRAPH REPRESENTATION AND TRAVERSAL ALGORITHMS**	**9 Hours**

Introduction to Graphs: Adjacency list and adjacency matrix representation. Graph Traversal Techniques: Depth-First Search (DFS) implementation, Breadth-First Search (BFS) implementation.	
TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	2	-	2	-	3	-	-	-	-	-	-	-	2	-	-
CO3	2	-	3	-	2	-	-	-	-	-	-	-	2	-	-
CO4	2	-	2	-	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	2	-	-	-	-	-	-	-	-	-	-	-	-
CO6	2	-	3	-	-	-	-	-	-	-	-	-	2	-	-
	2.17		2.4		2.5								2		

Books:

1. Seymour Lipschutz, Data Structures, Revised First Edition, McGraw-Hill Education.
2. Aaron M. Tenenbaum, Data Structures Using C, Prentice Hall.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, Eastern Economy Edition, PHI Learning Pvt. Ltd., 2010.
4. Donald Knuth, Art of Computer Programming, The: Volume 1: Fundamental Algorithms, Addison-Wesley, 1997.

Career Advancement & Skill Development-III (TIU-UEN-S297)

Program: B.Tech CSE	**Year, Semester: 2nd year, 3rd Sem**
Course Title: CAREER ADVANCEMENT & SKILL DEVELOPMENT-III	**Subject Code:** TIU-UEN-S297
Contact Hours/Week: 2-0-0 (L-T-P)	**Credit:** 2

COURSE OBJECTIVE:

Enable the student to:

1. Acquire basic communication skills in French.
2. Develop listening, speaking, reading, and writing abilities at a beginner level.
3. Understand and use simple grammatical structures and everyday vocabulary.
4. Engage in basic conversations in French related to common situations.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Recognize and use common French greetings and expressions.	K1
CO-2:	Memorize and repeat simple sentences using regular verbs and basic vocabulary.	K1
CO-3:	Understand and respond to basic questions about personal identity.	K2
CO-4:	Identify and explain short passages related to daily life.	K2
CO-5:	Construct short texts such as self-introductions and informal messages.	K3
CO-6:	Arrange isolated sentences and questions to engage in simple spoken exchanges in a variety of familiar contexts.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO FRENCH LANGUAGE**	**6 Hours**
The French alphabet and pronunciation, Greetings and introductions, Numbers and basic expressions of time		
MODULE 2:	**IDENTITY AND PERSONAL INFORMATION**	**6 Hours**
Talking about oneself and others, Nationalities, professions, and family, Using "être" and "avoir" verbs		
MODULE 3:	**EVERYDAY INTERACTIONS**	**6 Hours**
Asking for and giving personal details, Talking about preferences and habits, Introduction to regular -ER verbs		
MODULE 4:	**NAVIGATING DAILY SITUATIONS**	**6 Hours**
Ordering at a café or restaurant, Asking for directions, Using "aller" and "faire" verbs		
MODULE 5:	**DESCRIBING DAILY LIFE**	**6 Hours**
Talking about routines and leisure activities, Expressing likes and dislikes, Introduction to present tense conjugation		
	TOTAL LECTURES	**30 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	-	-	-	2	3	-	-	-	-	-
CO2	-	-	-	-	-	-	-	-	-	3	-	2	-	-	-
CO3	-	-	-	-	-	-	-	-	2	3	-	-	-	-	-
CO4	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-
CO5	-	-	-	-	-	-	-	-	-	3	-	2	-	-	-
CO6	-	-	-	-	-	-	-	-	2	3	-	2	-	-	-
									2	3		2			

Books:

Tech French - French for Science and Technology, Goyal Publishers, 2011

SEMESTER 4

Mathematics IV (TIU-UMA-T206)

Program: B. Tech. in CSE	**Year, Semester:** 2nd Yr., 4th Sem.
Course Title: Mathematics IV	**Subject Code:** TIU-UMA-T206
Contact Hours/Week: 3–1–0 (L–T–P)	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. learn the concept of probability, random variables, probability distribution.
2. obtain ideas about interpolation, numerical integration, and differentiation.
3. know the methods for solving simultaneous linear algebraic equations and differential equations.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	apply concept of probability and calculate the probability of random events	K4
CO-2:	apply probability distributions to solve simple problems	K3
CO-3:	understand the concept of error in numerical analysis	K2
CO-4:	develop an idea about different interpolation methods	K4
CO-5:	establish the concepts of numerical differentiation and integration	K4
CO-6:	apply various numerical methods for solving different types of problem such as simultaneous equations, algebraic, transcendental, and differential equations.	K4

COURSE CONTENT:

MODULE 1:	**PROBABILITY**	**20 Hours**
Classical, relative frequency and axiomatic definitions of probability, mutually exclusive events, independent events, conditional probability, Bayes' Theorem. Random Variables - Discrete and continuous random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments. Distributions - Uniform, Binomial, Geometric, Poisson, Exponential, Gamma, Normal distributions. Functions of a random variable.		
MODULE 2:	**NUMERICAL TECHNIQUES**	**25 Hours**
Approximations and round off errors, Truncation errors and Taylor Series. Interpolation – Newton's Forward, Backward, Lagrange Interpolation methods Numerical Differentiation, Numerical Integration – Trapezoidal, Simpson's 1/3rd rules Determination of roots of polynomials and transcendental equations by Bisection, Iteration, Newton-Raphson, Regula-Falsi methods. Solutions of linear simultaneous linear algebraic equations by Gauss Elimination and Gauss-		

Seidel iteration methods. Numerical solution of initial value problems by Euler, Modified Euler, Runge-Kutta 4th order.
TOTAL LECTURES **45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	-	-	-
CO2	3	2	-	-	1	-	-	-	-	-	-	-	-	-	-
CO3	3	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO4	2	-	2	-	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	2	-	-	-	-	-	-	-	-	-	-	-	-
CO6	3	2	2	-	1	-	-	-	-	-	-	-	2	-	-
	2.66	2	2		1.33								2		

Books:

1. Erwin Kreyszic-Advanced Engg. Mathematics
2. S. S. Sastry-An Introduction to Numerical Analysis.
3. Dutta and Jana- Numerical Analysis.
4. S. A. Mollah- Numerical Analysis and Computational Procedures
5. Probability and Statistics for Scientists and Engineers by Sheldon Ross

Computer Architecture (TIU-UCS-T220)

Program: B.Tech in CSE	**Year, Semester:** 2nd.,4th
Course Title: Computer Architecture	**Subject Code:** TIU-UCS-T220
Contact Hours/Week: 3-0-0 (L-T-P)	**Credit:** 3

COURSE OBJECTIVES:

1. To conceptualize the basics of organizational and architectural issues of a digital computer.
2. To analyze performance issues in processor and memory design of a digital computer.
3. To understand mapping techniques of different memory organization in digital computer.
4. To analyze processor performance improvement using instruction level parallelism.

COURSE OUTCOMES:

The students will be able to:

CO 1:	Describe the fundamental working principles of Basic Computer System architecture.	K 1
CO 2:	Identify the design of Pipeline Architecture.	K 2
CO 3:	Explain the memory hierarchy design and perform memory mapping operations.	K

		3
CO 4:	Identify the design of a pipelined CPU and hypothetical parallel processor architecture. VLIW and Superscalar Architecture.	K 4
CO 5:	Explain the hardware design of multiprocessors systems and its CPU-Memory interconnection architecture.	K 2
CO 6:	Non-Von Neumann Architecture.	K 2

COURSE CONTENT:

MODULE 1:	**Overview of von Neumann architecture**	**5 Hours**
Instruction set architecture; The Arithmetic and Logic Unit, The Control Unit, Memory and I/O devices and their interfacing to the CPU; Measuring and reporting performance; CISC and RISC processors.		
MODULE 2:	**Pipelining**	**12 Hours**
Basic concepts of pipelining, data hazards, control hazards, and structural hazards; Techniques for overcoming or reducing the effects of various hazards.		
MODULE 3:	**Hierarchical Memory Technology**	**10 Hours**
Inclusion, Coherence and locality properties; Cache memory organizations, Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies.		
MODULE 4:	**Instruction-level parallelism**	**10 Hours**
Concepts of instruction-level parallelism (ILP), Techniques for increasing ILP; Superscalar, super-pipelined and VLIW processor architectures; Vector and Array processors.		
MODULE 5:	**Multiprocessor Architecture**	**6 Hours**
Taxonomy of parallel architectures; Centralized shared-memory architecture, synchronization, memory consistency, interconnection networks; Distributed shared-memory architecture, Cluster computers.		
MODULE 6:	**Non von Neumann Architectures**	**2 Hours**
Data flow Computers, Reduction computer architectures, Systolic Architectures.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-

CO2	3	2	2	-	-	-	-	-	-	-	-	-	2	-	-
CO3	3	2	-	-	1	-	-	-	-	-	-	-	2	-	-
CO4	3	-	2	-	1	-	-	-	-	-	-	-	2	-	-
CO5	3	2	2	-	-	-	-	-	-	-	-	-	2	-	-
CO6	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
	3	2	2		1								2		

Books:

1. M Morris Mano, Computer System Architecture (Revised 3rd Edition), Pearson.
2. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann.
3. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill.
4. John Paul Shen and Mikko H. Lipasti, Modern Processor Design: Fundamentals of Superscalar Processors, Tata McGraw-Hill.
5. M. J. Flynn, Computer Architecture: Pipelined and Parallel Processor Design, Narosa Publishing House.

Graph Theory and Combinatorics (TIU-UCS-T224)

Program: B. Tech. in CSE	**Year, Semester:** 4th., 7th
Course Title: Graph Theory and Combinatorics	**Subject Code:** TIU-UCS-T224
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Introduce students to the fundamental concepts of graph theory, including different types of graphs, graph representations, and essential properties.
2. Enable students to identify and mathematically formulate problems related to graph theory.
3. Teach students to analyze and apply graph-theoretic algorithms for solving problems.
4. Develop an understanding of combinatorial principles, including permutations, combinations, and their applications in problem-solving.

COURSE OUTCOME:

The students will be able to:

CO1:	Describe the concepts of graphs and related tools and techniques.	K2
CO2:	Identify and formulate problems which are related to graph theory and combinatorics.	K3
CO3:	Analyze problems and apply graph-theoretic techniques for problem-solving.	K3

CO4:	Describe the principles of combinatorics and apply them for problem-solving.	K3
CO5:	Perform combinatorial analysis of problems of diverse nature.	K4
CO6:	Develop efficient algorithms and models using graph theory and combinatorial techniques to solve real-world problems.	K4

COURSE CONTENT :

MODULE 1:	**FUNDAMENTALS OF GRAPH THEORY**	**9 Hours**
Binary relation on set, concepts and definition of graph; undirected, directed, weighted graphs, and simple graphs; complete graphs, sparse graphs, and dense graphs. adjacency of vertices and degree of vertices; relationship between number of vertices and number of edges; incidence between vertices and edges.		
MODULE 2:	**GRAPH ALGORITHMS AND APPLICATIONS**	**10 Hours**
Operations on graph; graph isomorphism; Path connectivity and connected components in a graph, properties and computation. Euler's Tour, Hamiltonian path, Topological sorting, Traveling salesman's problem. strongly connected graph and components. representations of a graph in computer; graph exploration techniques: Breadth-first search (BFS) and Depth-first search (DFS) and their applications.		
MODULE 3:	**SPANNING TREES, SHORTEST PATHS, AND GRAPH PROPERTIES**	**9 Hours**
Spanning tree of a graph, minimum spanning tree (MST) of a weighted graph, its properties and computation (Kruskal's and Prim's algorithms). Concepts of shorted paths in a graph, their properties and computation. Concepts of planar graphs and their properties; dual graph of a planar graph; bipartite graphs; tree as a graph and its properties; graph coloring.		
MODULE 4:	**COMBINATORICS AND COUNTING PRINCIPLES**	**9 Hours**
Recapitulation of concepts from previous discrete mathematics course; the rules of sum and product; permutations, combinations, selection; The pigeonhole principle and its variants; The Binomial theorem, combinations with repetition, the Catalon numbers; The multinomial theorem. Combinatorial problems in discrete probability.		
MODULE 5:	**ADVANCED COUNTING TECHNIQUES AND RECURRENCE RELATIONS**	**6 Hours**
Principles of inclusion and exclusion, and its variants; derangements; generating functions; recurrence relations. Problem solving with these principles.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-	-	2	-	-
CO3	3	2	2	-	-	-	-	-	-	-	-	-	2	-	-
CO4	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO5	2	2	2	-	-	-	-	-	-	-	-	-	2	-	-
CO6	3	3	3	-	1	-	-	-	-	-	-	-	2	2	2
	2.83	2.33	2.33		1								2	2	2

Books:

1. "Graph Theory with Applications" – J.A. Bondy and U.S.R. Murty
2. "Introduction to Graph Theory" – Douglas B. West
3. "Graph Theory" – Reinhard Diestel.
4. "Graphs, Networks, and Algorithms" – Dieter Jungnickel

Object Oriented Programming (TIU-UCS-T214)

Program: B. Tech. in CSE	**Year, Semester:** 2nd, 4th
Course Title: Object Oriented Programming	**Subject Code:** TIU-UCS-T214
Contact Hours/Week: 3–0–2 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. understand the real-world problem and design solutions by object-oriented programming
2. analyze the nature of problems solved with object-oriented techniques
3. design and implement suitable programming approach for a given application

COURSE OUTCOME:

The students will be able to:

CO-1:	Define an object-oriented approach to programming and identify potential benefits of object-oriented programming over other approaches	K2
CO-2	Understand the difference between the top-down and bottom-up program design approach	K2
CO-3:	Demonstrate the use of various OOPs concepts using C++	K3
CO-4:	Solving a computational problem by implementing the solution as a real-world entity	K4
CO-5:	Understand and apply some advanced constructs of C++ like virtual functions, operator overloading, exception handling, standard template library	K3
CO-6:	Analyze and optimize C++ programs for efficiency, maintainability, and scalability using OOP best practices.	K4

COURSE CONTENT:

MODULE 1:	INTRODUCTION	10 Hours
Introduction to Object Oriented Paradigm, Need of object-oriented design, Drawbacks of Procedure Oriented Programming, Features of object-oriented languages, POP Vs OOP, Benefits & Applications of OOP, Difference between C and C++.		
MODULE 2:	**BASIC CONCEPTS OF OBJECT ORIENTATION**	**12 Hours**
Class, Object, Data abstraction, Encapsulation, Inheritance, Polymorphism, Message Passing, Dynamic binding.		
MODULE 3:	**FUNDAMENTALS OF OOPs**	**12 Hours**
Fundamentals of OOPs: Class & Objects, Constructors & Destructor. Different perspectives on inheritance, Types of inheritance, Polymorphism: Compile Time & Run time Polymorphism, Virtual functions, Virtual table construction, Overloading, Overriding, Abstract Class, Virtual Class.		
MODULE 4:	**ADVANCE OOP CONCEPTS**	**11 Hours**
Class and Function Templates, Standard Template Libraries in C++: lists, vectors, sets, maps. Exceptions Handling.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	2	3	-	-	-	-	-	-	-	-	-	-	2	-	-
CO3	3	2	2	-	2	-	-	-	-	-	-	-	3	-	-
CO4	3	2	3	-	-	-	-	-	-	-	-	-	3	-	-
CO5	3	-	2	-	2	-	-	-	-	-	-	-	3	-	-
CO6	3	2	3	-	2	-	-	-	-	-	-	1	3	-	2
	2.83	2.2	2.5		2							1	2.66		2

Books:

1. Robert Lafore, Object-Oriented Programming in C++, Fourth Edition, Pearson.
2. Herbert Schildt, C++: The Complete Reference, Fourth Edition, Mc-Graw Hill Education, India, 2003
3. Bjarne Stroustrup, The C++ Programming Language, Third Edition, Pearson, 2000.
4. E. Balagurusamy, Object-Oriented Programming with C++, 8th Edition, Mc-Graw Hill Education 2021
5. Scott Meyers, Effective Modern C++, O'Reilly Media, Inc., 2014.
6. Scott Meyers, Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library, Addison-Wesley Professional Computing Series,2001.

Microprocessor & Microcontroller (TIU-UEC-T210)

Program: B. Tech. in CSE	**Year, Semester:** 2nd, 4th
Course Title: Microprocessor & Microcontroller	**Subject Code:** TIU-UEC-T210
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Learn about the fundamentals of Microprocessor and Microcontroller Systems and its Components. They should be able to write and understand the assembly language program by applying the concept of 8085 and 8086 microprocessors' architecture, interrupts, instructions sets etc.
2. Understand understand the working principle, pin diagram, memory interfacing, instructions of 8051 microcontroller so that they can design application specific systems.
3. Design and analyse the microprocessor based system they need to understand the operations of different peripheral devices such as 8255, 8257/8237A etc. and corresponding interfacing circuits.
4. Understand measurement and control techniques of different electrical, physical quantities by designing and analysing processor based systems.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Understand the fundamentals of microprocessors and microcontrollers	K2
CO-2:	Develop assembly language programming skills for 8085 microprocessor	K2
CO-3:	Analyze and design interfacing techniques for peripheral devices	K4
CO-4:	Demonstrate knowledge of 8086 microprocessor architecture and programming	K2
CO-5:	Demonstrate knowledge of 8051 microcontroller architecture and programming	K2
CO-6:	Apply microprocessor and microcontroller concepts to measurement and control applications	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO 8085 (8 BIT PROCESSOR)**	**15 Hours**
Introduction to 8085 (8 bit processor): Evolution of microprocessors and microcontrollers, CPU architecture-register organization, pin description and features, addressing modes, Instruction set and Assembly Language Programming. Instruction cycle, machine cycle, Timing diagram, stack and subroutine operation.		
MODULE 2:	**HARDWARE INTERFACING AND PERIPHERALS**	**10 Hours**
Hardware Interfacing and Peripherals: Interfacing memory, IO devices (IO mapped IO& Memory mapped IO). Interrupts, Interfacing 8255 peripheral, DMA controller (8237).		

MODULE 3:	**INTRODUCTION TO 8086 (16 BIT PROCESSOR)**	**10 Hours**
Introduction to 8086 (16 bit processor): Architecture, register organization, pin description and features. Instruction Set and Addressing modes, interfacing memory, IO devices, Interrupts.		
MODULE 4:	**INTRODUCTION TO 8051 (8 BIT MICROCONTROLLER)**	**10 Hours**
Introduction to 8051 (8 bit microcontroller): Architecture, Pin description and features, Special function registers, I/O pin ports and circuits, Internal, External memories, addressing modes, Assembly Language Programming.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO3	3	2	3	-	2	-	-	-	-	-	-	-	3	-	2
CO4	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO5	3	-	2	-	2	-	-	-	-	-	-	-	2	-	2
CO6	3	2	3	-	2	-	-	-	-	-	-	1	3	-	3
	3	2	2.4		2							1	2.33		2.33

Books:

1. R. Gaonkar, “Microprocessor Architecture, Programming, and Applications with the 8085”, Penram.
2. B.Ram, “Fundamentals of Microprocessors and Microcontrollers” , Dhanpat Rai Publication.
3. D. V. Hall, “Microprocessors and Interfacing”, McGraw Hill
4. K. M. Bhurchandi and A. K. Ray, “Advanced Microprocessors and Peripherals”, Tata McGraw Hill
5. M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, “The 8051 Microcontroller and Embedded Systems”, Pearson
6. K. Ayala, “The 8051 Microcontroller”, Delmar Cengage Learning

Numerical Lab (TIU-UCS-L216)

Program: B.Tech. in CSE	**Year, Semester:** 2nd, 4th.
Course Title: Numerical Lab	**Subject Code:** TIU-UCS-L216
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. apply numerical techniques to approximate functions and find the area under them
2. use numerical methods to find approximate roots of equation, find solutions to systems of equations, solve differential equations etc.
3. use computational tools to implement the numerical methods

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1:	approximate a function by a linear polynomial	K4
CO-2:	calculate the approximate the area under a curve	K4
CO-3:	use numerical techniques to find the approximate solution of alegebraic and transcendental equations	K4
CO-4:	find the exact solution of a system of equations by direct methods	K4
CO-5:	find the approximate solution of a system of equations by indirect methods	K4
CO-6:	find the numerical solution for initial value problems (IVPs)	K4

COURSE CONTENT:

MODULE 1:	**FORWARD AND BACKWARD DIFFERENCE TABLE**	**9 Hours**
Introduction to finite differences and their applications, Forward and Backward Difference Operators, Construction of Difference Tables, Higher-order differences and accuracy analysis		
MODULE 2:	**INTERPOLATION METHODS**	**9 Hours**
Newton's Forward and Backward Interpolation Formulas, Lagrange's Interpolation Method, Applications of interpolation in numerical computations		
MODULE 3:	**NUMERICAL INTEGRATION**	**6 Hours**
Introduction to Numerical Integration, Trapezoidal Rule and its implementation, Simpson's 1/3rd Rule and applications, Error analysis in numerical integration		
MODULE 4:	**NUMERICAL SOLUTION OF ALGEBRAIC AND TRANSCENDENTAL EQUATIONS**	**6 Hours**
Bisection Method: Concept, Algorithm, and Implementation, Newton-Raphson Method: Convergence, Limitations, and Applications		
MODULE 5:	**NUMERICAL SOLUTION OF A SYSTEM OF EQUATIONS**	**6 Hours**
Introduction to solving systems of linear equations, Gauss Elimination Method: Direct approach to		

solving equations, Gauss-Seidel Iterative Method: Convergence and Stopping Criteria		
MODULE 6:	**NUMERICAL SOLUTION OF INITIAL VALUE PROBLEMS**	**9 Hours**
Euler's Method: Concept and Implementation, Modified Euler's Method for higher accuracy, Runge-Kutta Methods: 2nd and 4th Order Techniques, Applications in solving Ordinary Differential Equations (ODEs).		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO3	3	2	3	-	2	-	-	-	-	-	-	-	3	-	2
CO4	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO5	3	-	2	-	2	-	-	-	-	-	-	-	2	-	2
CO6	3	2	3	-	2	-	-	-	-	-	-	1	3	-	3
	3	2	2.4		2							1	2.33		2.33

Books:

1. Balagurusamy, E. (2017). Numerical Methods (1st ed.). McGraw-Hill Education.
2. Veerarajan, T., & Ramachandran, T. (2006). Numerical Methods with Programs in C and C++ (1st ed.). Tata McGraw-Hill.
3. Chapra, S. C. (2018). Applied Numerical Methods with MATLAB for Engineers and Scientists (4th ed.). McGraw-Hill Education.
4. Pradeep, N., & Govindarajan, G. (2008). Numerical Methods and Computer Programming (1st ed.). New Age International Publishers.
5. Grewal, B. S. (2019). Numerical Methods in Engineering and Science with Programs in C and C++ (10th ed.). Khanna Publishers.
6. Rajaraman, V. (2012). Computer Oriented Numerical Methods (3rd ed.). PHI Learning Pvt. Ltd.

Computer Architecture Lab (TIU-UCS-L220)

Program: B.Tech. in CSE	**Year, Semester: 2**nd, 4th.
Course Title: Computer Architecture Lab	**Subject Code:** TIU-UCS-L220
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand the architectural principles of computer systems and their organization.
2. Implement and analyze fundamental hardware components.
3. Develop memory and arithmetic logic unit (ALU) designs for computing applications.

COURSE OUTCOME:

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

CO-1	Explain the fundamental aspects of computer architecture design and analysis.	K3
CO-2	Demonstrate a comprehensive understanding of basic hardware concepts with Integrated Circuits	K3
CO-3	Illustrate the operations of different gates and their applications	K3
CO-4	Implement hardware designs and apply them in various domains.	K4
CO-5	Design the memory unit of a computer system.	K4
CO-6	Develop the arithmetic logic unit of a computer system.	K3

COURSE CONTENT:

MODULE 1:	**BASIC LOGIC FUNDAMENTALS**	**9 Hours**
Boolean algebra, Logic gates, Truth tables, K-map simplification, Implementation of logic functions using gates		
MODULE 2:	**COMBINATIONAL CIRCUIT DESIGN**	**9 Hours**
Design and implementation of Adders, Subtractors, Multiplexers, Decoders, and Encoders using logic gates		
MODULE 3:	**SEQUENTIAL CIRCUITS AND STATE MACHINES**	**6 Hours**
Flip-Flops (SR, D, JK, T), Registers, Counters (Synchronous and Asynchronous), Finite State Machines		
MODULE 4:	**MEMORY ORGANIZATION AND ADDRESSING**	**6 Hours**
RAM and ROM architectures, Cache memory design, Memory hierarchy, Address decoding		
MODULE 5:	**ARITHMETIC LOGIC UNIT (ALU) AND PROCESSOR DESIGN**	**6 Hours**
Design of an ALU, Arithmetic operations (addition, subtraction, multiplication), Logical operations, Bitwise manipulations		
MODULE 6:	**CPU DESIGN AND PERFORMANCE OPTIMIZATION**	**9 Hours**
Instruction set architecture, Execution cycle, Control unit design, Pipelining, Performance		

evaluation metrics	
TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO2	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO3	3	2	3	-	2	-	-	-	-	-	-	-	3	-	2
CO4	3	2	2	-	2	-	-	-	-	-	-	-	2	-	-
CO5	3	-	2	-	2	-	-	-	-	-	-	-	2	-	2
CO6	3	2	3	-	2	-	-	-	-	-	-	1	3	-	3
	3	2	2.4		2							1	2.33		2.33

Books:

1. M. Morris Mano & Michael D. Ciletti - Digital Design (5th ed.), Pearson
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky - Computer Organization and Embedded Systems (6th ed.), McGraw-Hill
3. William Stallings - Computer Organization and Architecture (10th ed.), Pearson
4. David A. Patterson & John L. Hennessy - Computer Organization and Design (5th ed.), Elsevier

Object Oriented Programming Lab (TIU-UCS-L214)

Program: B.Tech. in CSE	**Year, Semester:** 2nd, 4th.
Course Title: Object Oriented Programming Lab	**Subject Code:** TIU-UCS-L214
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand the mathematical foundations of various numerical methods and their significance in solving engineering and scientific problems.
2. Apply numerical techniques to solve algebraic equations, system equations, interpolation, differentiation, and integration using computational tools.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Able to understand the foundational principles of object-oriented programming and its application in solving computational problems using C++.	K2
CO-2	Able to comprehend and construct algorithms and flowcharts for problem-solving in object-oriented programming contexts.	K3
CO-3	Able to apply object-oriented programming concepts like classes, inheritance, and polymorphism in software development using C++.	K3
CO-4	Able to design, implement, and test object-oriented programs to solve real-world scenarios using appropriate programming techniques.	K4
CO-5	Able to analyze and optimize object-oriented solutions for complex problems encountered in industrial applications and software development projects.	K4
CO-6	Evaluate and debug object-oriented programs to ensure robustness, efficiency, and adherence to best practices in software development using C++.	K4

COURSE CONTENT :

MODULE 1:	**C++ BASICS AND STRUCTURES**	**4 Hours**
Introduction to C++, Data types, Control Flow and loops, Mathematical Operations using loops and conditions, Introduction of Structure in C++, Implementing Structure in Real World Scenarios, Using Structure for practical applications		
MODULE 2:	**CLASSES AND OBJECTS, DATA STRUCTURES USING OBJECTS**	**7 Hours**
Introduction to Classes and Objects, Object-Oriented Program Design, Real-World Example: Toy-Train Ticket Counter, Implementing a Vector Class, Introduction to Data Structures Using Objects, Implementing a Linked List and Stack, Practical Application and Optimization		
MODULE 3:	**CONSTRUCTORS, FUNCTIONS, ENUM AND STRING**	**9 Hours**
Introduction to Constructors and Destructors, different types of Constructors, Introduction to function, Different use cases of Function implementation, Enum and String, String Handling in C++		
MODULE 4:	**FRIEND FUNCTION, INLINE FUNCTION, POLYMORPHISM, VIRTUAL FUNCTION**	**9 Hours**
Implementation of Friend class and Friend Function, Introduction to Inline Function with implementation of inline mathematical operations, Implementation of Polymorphism, Run time and Compile time polymorphism, Function Overloading, Operator Overloading, Function Overriding, Runtime Polymorphism using Virtual Functions		
MODULE 5:	**INHERITANCE, ABSTRACT CLASSES, AND FILE HANDLING**	**8 Hours**
Implementation of Inheritance, Different types of Inheritance and its use cases,Abstract Classes and Pure Virtual Functions, Introduction to File Handling in C++, Reading and Writing Files		
MODULE 6:	**TEMPLATES, EXCEPTIONS, STANDARD TEMPLATE LIBRARY(STL) AND MULTI-FILE PROGRAMS: MINI PROJECT**	**8 Hours**
Introduction to Templates -Concept of generic programming, Function templates, Class templates, Exception Handling, Standard Template Library (STL),		

Introduction to Multi-File Programs and implementing a Mini Project	
TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	3	-	-	-	-	-	-	-	2	-	-
CO2	2	2	3	-	3	-	-	-	2	2	-	-	2	-	-
CO3	2	3	3	-	3	-	-	-	2	2	-	-	3	-	2
CO4	3	2	2	-	3	-	-	-	2	2	-	-	2	-	-
CO5	2	-	3	-	3	-	-	-	2	2	-	-	2	-	2
CO6	3	2	3	-	3	-	-	-	2	2	-	1	3	-	3
	2.5	2.2	2.8		3				2	2		1	2.33		2.33

Books:

1. *Bjarne Stroustrup(2013).* The C++ Programming Language(4th ed.) Addison-Wesley.
2. Herbert Schildt (2014). *C++: The Complete Reference* (4th ed.). McGraw-Hill.
3. *Bjarne Stroustrup(2014).* Programming: Principles and Practice Using C++(2nd Ed.) Addison-Wesley.
4. Paul Deitel & Harvey Deitel(2016). C++ How to Program(10th ed.). Pearson.
5. Stanley B. Lippman, Josée Lajoie, & Barbara E. Moo (2012). *C++ Primer* (5th ed.). Addison-Wesley.
6. Matt Weisfeld (2019). *The Object-Oriented Thought Process* (5th ed.). Addison-Wesley.

Microprocessor and Microcontroller Lab (TIU-UEC-L218)

Program: B.Tech. in CSE	**Year, Semester:** 2nd, 4th.
Course Title: Microprocessor and Microcontroller Lab	**Subject Code:** TIU-UEC-L218
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

5. Learn how to write, execute and debugassemble language programs (ALPs) for microprocessors.
6. Understand data manipulation, arithmetic, bitwise and logical operations for microprocessors.
7. Understand about memory addressing and data storage for microprocessors.

8. Understand the uses of looping, branching and conditional and unconditional jumps in ALPs to control execution flow for microprocessors.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Understand microprocessor architecture, memory organization, and register operations and assembly language programming.	K2
CO-2	Perform data manipulation and bitwise operations for complement, shifting, masking, and mask-off of data.	K3
CO-3	Implement arithmetic and logical operations to perform addition, subtraction, multiplication and division.	K3
CO-4	Apply data conversion and encoding techniques for efficient data representation.	K3
CO-5	Develop and implement sorting and searching algorithms.	K4
CO-6	Learn how to store and retrieve data from specific memory locations, and debug and optimize assembly language programs for efficient execution.	K4

COURSE CONTENT:

MODULE 1:	**DATA MANIPULATION AND BITWISE OPERATIONS**	**6 Hours**
1's and 2's complement of input data, shifting or rotating of accumulator contents.		
MODULE 2:	**ARITHMETIC OPERATIONS**	**9 Hours**
Addition and subtraction of input data using control flow (loop), use of different arithmetic and branch control instructions.		
MODULE 3:	**DATA ENCODING AND MASKING**	**6 Hours**
Pack (mask) and unpack (mask off) of input data, checking even or odd input data, use of logical and branch control instructions.		
MODULE 4:	**DATA CONVERSION**	**9 Hours**
BCD to Binary conversion and vice-versa: use of arithmetic, logical and branch control instructions.		
MODULE 5:	**SORTING AND SEARCHING ALGORITHMS**	**9 Hours**
Searching and sorting of data from an array: handling of multiple memory locations for different data, use of branch control instructions.		
MODULE 6:	**ADVANCED ARITHMETIC OPERATIONS**	**9 Hours**
Multiplication and division of input data using control flow (loop), use of different arithmetic and branch control instructions, repetitive addition and subtraction methods.		
TOTAL LAB HOURS		**48 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	-	-	3	-	-	-	-	-	-	-	2	-	-
CO2	3	2	3	-	3	-	-	-	2	2	-	-	2	-	-

CO3	3	2	3	-	3	-	-	-	2	2	-	-	3	-	2
CO4	3	2	2	-	3	-	-	-	2	2	-	-	2	-	-
CO5	3	2	3	-	3	-	-	-	2	2	-	-	2	-	2
CO6	3	2	3	-	3	-	-	-	2	2	-	1	3	-	3
	3	2	2.8		3				2	2		1	2.33		2.33

Books:

5. R. Gaonkar, "Microprocessor Architecture, Programming, and Applications with the 8085",Penram
6. A. Nagoor Kani,"Microprocessor (8085) And its Applications" 2005, McGraw-Hill Education
7. D. V. Hall, "Microprocessors and Interfacing", McGraw Hill
8. M. A. Mazidi, J. G. Mazidi and R. D. McKinlay, "The 8051 Microcontroller and Embedded Systems", Pearson

Career Advancement & Skill Development-IV-Prompt Engineering

Program: B. Tech. in CSE	**Year, Semester: 2ND** , 4th
Course Title: Prompt Engineering	**Subject Code:** TIU-CASD-UCS-S296A
Contact Hours/Week: 0–0–2 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. to understand the fundamentals of prompt engineering
2. Master the art of crafting effective prompts for various AI models
3. Explore various prompting techniques to achieve desired outcomes from generative AI
4. Understand the ethical concerns and know the ways to mitigate the associated risks

COURSE OUTCOME:

The student will be able to:

CO1:	Explain the theoretical foundations of prompt engineering and its evolving significance in LLMs	K1
CO2:	Analyze various prompting techniques and assess their effectiveness using theoretical models.	K3
CO3:	Investigate advanced prompt structures and their influence on model behavior.	K3
CO4:	Evaluate LLM performance and limitations across different prompt scenarios.	K4

CO5:	Identify and mitigate ethical concerns, biases, and constraints in prompt design.	K2
CO6:	Implement theoretical principles of prompt engineering in practical applications.	K3

COURSE CONTENT:

MODULE 1:	**Foundations of Large Language Models and Prompt Engineering**	**8 Hours**
Historical perspective on NLP and the rise of LLMs. Theoretical understanding of LLM architecture, including Transformer models. Role of tokenization, embeddings, and attention mechanisms in LLMs. Introduction to prompt engineering: theoretical foundation, purpose, and limitations. Exploration of prompt engineering as a bridge between NLP tasks and LLM capability		
MODULE 2:	**Theoretical Models of Prompting and Prompt Typology**	**8 Hours**
Classification and analysis of different types of prompts (direct, indirect, zero-shot, few-shot). Theoretical underpinnings of zero-shot and few-shot learning in LLMs. Prompt-based learning theory: how LLMs interpret and respond to structured input. Analysis of prompt effectiveness through probabilistic and statistical models. The concept of prompt transferability and adaptability across tasks		
MODULE 3:	**Principles of Prompt Design and Evaluation**	**5 Hours**
Theoretical principles for effective prompt construction (clarity, specificity, contextual relevance). Methods to optimize prompt structures for model coherence and reliability. Analytical frameworks for assessing prompt quality and model interpretability. The role of heuristics and biases in human prompt design. Understanding the relationship between prompt variability and output diversity		
MODULE 4:	**The Impact of Prompt Structure on Model Behavior**	**8 Hours**
Theoretical exploration of prompt-induced bias and model behavior manipulation. Analysis of prompt chaining, task decomposition, and control prompts. Understanding model interpretability: How LLMs respond to and process varied prompts. Theories of response consistency, coherence, and fluency in model output. Introduction to reinforcement learning as a method for optimizing prompt structure		
MODULE 5:	**Applications and Domain-Specific Theories of Prompt Engineering**	**5 Hours**
Overview of domain-specific prompt engineering applications: legal, medical, creative industries, etc. Analytical perspectives on prompt adaptability in specialized fields. Limitations of LLMs in domain-specific tasks and ways to overcome these through prompt design. Domain-specific prompt challenges: specificity, jargon, and context adaptation. Review of case studies where prompt engineering contributed to success in critical domains		
MODULE 6:	**Ethical, Philosophical, and Social Implications of Prompt Engineering**	**6 Hours**
Ethical theories and frameworks as applied to AI and LLM-driven prompt engineering. Theoretical discussion on biases in LLM outputs and prompt-related ethical dilemmas. Philosophical questions on language, meaning, and intent in AI-generated content. Regulatory and ethical guidelines for responsible prompt engineering. Future directions in prompt engineering: ethical considerations in an evolving field		
MODULE 7:	**Project-Based Learning with Theoretical Applications**	**5 Hours**
Synthesis of theoretical principles in real-world prompt engineering applications. Project-based exploration of prompt engineering in novel applications. Comparative analysis of theoretical models vs. practical outcomes in prompt engineering. Capstone project focused on domain-specific prompt design, tuning, and evaluation		

TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	1	-	-	-	-	-	-	2	3	3	3
CO2	2	2	2	2	2	-	-	-	-	-	-	1	3	3	3
CO3	3	3	1	2	2	-	-	-	-	-	-	1	3	3	3
CO4	2	2	-	2	3	-	-	-	-	-	-	1	3	3	3
CO5	1	1	-	-	1	-	-	-	2	3	-	1	2	2	2
CO6	2	-	-	-	-	2	-	2	-	2	-	1	2	2	2
	2.166	2	1.5	2	1.8	2		2	2	2.5		1.167	2.667	2.667	2.667

Books:

1. Nathan Hunter, “The Art of Prompt Engineering with Chatgpt: A Hands-On Guide: 3 (Learn AI Tools the Fun Way!)” 2023, ISBN: 1739296710, ISBN-13: 978-1739296711.
2. James Phoenix, “Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs”, Eighth Edition (O’Reilly Media), 2024, ISBN-13: 9781098153434.

SEMESTER 5

Design and Analysis of Algorithm (TIU-UCS-T321)

Program: B. Tech. in CSE	**Year, Semester:** 3rd., 5th
Course Title: Design and Analysis of Algorithm	**Subject Code:** TIU-UCS-T321
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Understand the Fundamental Principles of Algorithm Design
2. Master Asymptotic Analysis and Notations
3. Analyze Algorithm Efficiency in Different Scenarios
4. Apply Algorithm Design to Real-world Problems.

COURSE OBJECTIVE:

The student will be able to:

CO1	Explain the basic concepts involved in designing, analyzing, and implementing algorithms.	K2
CO2	Analyze problem characteristics to devise efficient algorithms tailored to specific tasks.	K3
CO3	Identify and distinguish between tractable and intractable problems in algorithm design.	K3
CO4	Utilize algorithm design principles to solve real-world problems, implementing solutions and conducting complexity analyses	K4
CO5	Assess and compare the efficiency of various algorithms based on time and space complexity.	K3
CO6	Apply advanced algorithmic techniques, such as approximation and parallel algorithms, to solve complex problems efficiently	K3

COURSE CONTENT:

MODULE 1:	**Foundation of Algorithm & Analysis**	**10 Hours**
Introduction to algorithm design and importance of its analysis, Asymptotic notations and their significance, Complexity analysis of algorithms – best case, worst case and average case with example of Insertion sort, Quick sort and Heap sort, Time & space trade-offs, Analysis of recursive algorithms – Substitution method, Recursion tree method and Masters' theorem, Lower bound for comparison-based sort.		
MODULE 2:	**Algorithmic Paradigms**	**10 Hours**
Classification of algorithm design techniques for problem solving: Brute-force, Divide-and-		

Conquer, Greedy, Dynamic Programming, Backtracking and Branch-and-Bound, Methodology and application domains, Illustration of the techniques with suitable examples: Activity selection, Huffman code, Knapsack problem, Matrix Chain Multiplication, 8-Queen problem, 15-puzzle problem. [extra problem in tutorial]		
MODULE 3:	**Graph Algorithms**	**12 Hours**
Traversal algorithms: DFS, BFS - concept, complexity analysis and applications, Minimum Spanning Tree finding algorithm: Prim's, Kruskal - concept, complexity analysis, Disjoint set operations, shortest path finding algorithm: single source and all pairs –Bellman-Ford, Dijkstra and Floyd-Warshall, Topological sort, Network flow algorithm: Ford-Fulkerson, Max-flow Min-cut theorem.		
MODULE 4:	**Problem Reducibility and NP-completeness**	**8 Hours**
Problem classification on Computability: P, NP, NP-complete and NP-hard, Reducibility of NP-complete problems with example – Satisfiability, Vertex cover, Traveling Salesman problem, Cook's theorem.		
MODULE 5:	**Advanced Topics**	**5 Hours**
Approximation algorithm, Randomized algorithm technique Amortized analysis.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	1	2	-	-	-	-	-	-	2	3	-	-
CO2	3	3	2	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	2	3	2	-
CO4	3	3	3	2	-	-	-	-	-	-	-	2	3	3	-
CO5	3	3	2	3	-	-	-	-	-	-	-	2	3	3	-
CO6	3	3	3	2	-	-	-	-	-	-	-	2	3	3	-
	3	2.83	2.4	2	2							2	3	2.75	

Books:

1. Introduction to Algorithms- Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein, The MIT Press

2. Fundamentals of computer algorithms by Satroj Sahani and Ellis Horowitz.

Database Management System (TIU-UCS-T301)

Program: B. Tech. in CSE	**Year, Semester:** 3rd., 5th.
Course Title: Database Management System	**Subject Code:** TIU-UCS-T301
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

1. Understand the basic concepts and the applications of database systems and the relational database design principles.
2. Master the basics of SQL and construct queries using SQL.
3. Familiar with the basic issues of transaction processing and concurrency control and database storage structures and access techniques.

COURSE OUTCOME:

The students will be able to:

CO1:	Understand the core concepts and features of Database Management System	K2
CO2:	Design and development of DBMS solutions based on relevant project work	K3
CO3:	Analyze and troubleshoot database related problems and finding the solution using the DBMS knowledge as acquired	K4
CO4:	Study the latest trends in DBMS and get the connectivity with the cutting-edge technologies	K3
CO5:	Implement database security, backup, and recovery techniques to ensure data integrity.	K3
CO6:	optimize SQL queries and database operations for improved performance.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**2 Hours**
General introduction to database systems, Concept of file System and Disadvantages, Database-DBMS distinction, Role of DBA, Approaches to building a database, Data models, Database management system, Three-schema architecture of a database, Data Independency, Integrity constraints.		
MODULE 2:	**RELATIONAL DATA MODEL**	**2 Hours**
Concept of relations, Schema-instance distinction, Keys, Referential integrity and foreign keys. Relational Algebra Operators: Selection, Projection, Union, Intersection, Set difference, Cross product, Rename, Assignment, Various types of joins, Division, Example queries. Tuple Relational Calculus, Domain Relational Calculus.		
MODULE 3:	**SQL (STRUCTURED QUERY LANGUAGE)**	**7 Hours**
Introduction, Data definition in SQL, Table, key and foreign key definitions, Update behaviors, querying in SQL, Basic select- from- where block and its semantics, Nested queries-correlated and uncorrelated, Notion of aggregation, Aggregation functions group by and having clauses,		

Embedded SQL		
MODULE 4:	**DATABASE DESIGN CONCEPTS (PART-1)-DEPENDENCIES AND NORMAL FORMS**	**9 Hours**
Importance of a good schema design, Problems encountered with bad schema designs, Motivation for normal forms, dependency theory – functional dependencies, Armstrong's axioms for FD's, Closure of a set of FDs, Minimal covers, Definitions of 1NF, 2NF, 3NF and BCNF, Decompositions and desirable properties of them, Algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, Join dependencies and definition of 5NF, DKNF.		
MODULE 5:	**DATABASE DESIGN CONCEPTS (PART-2) -ER MODEL**	**6 Hours**
Conceptual data modeling-motivation, Entities, Entity types, Various types of attributes, Relationships, Relationship types, E/R diagram notation, High-level conceptual modeling, ER Modeling concepts, ER Diagrams, Cardinality constraints Enhanced ER Model: Higher-order relationships, Enhanced ER Model (EER), Weak-entity types, Subclasses and inheritance, Specialization and Generalization, Modeling of UNION types using categories.		
MODULE 6:	**DATA STORAGE AND INDEXES**	**7 Hours**
File organizations, Primary, Secondary index structures, Various index structures - hash-based, Dynamic hashing techniques, Multi-level indexes, B+ trees.		
MODULE 7:	**TRANSACTION PROCESSING AND CONCURRENCY CONTROL**	**9 Hours**
Transaction Fundamentals: OLTP environments, Concurrency issues, need for transactions, Necessary properties of transactions (ACID properties), Transaction states, serializability, Serial schedules, Conflict serializability, View serializability, Recoverable and non-recoverable schedules, Cascading rollbacks, Cascadeless schedules. Concurrency control: Serialized and non-serialized schedules, Testing for serializability, Locking, Lock compatibility matrix, Locking and serializability, Deadlocks and starvation, Two-phase locking (2PL) protocol, Conservative, strict and rigorous 2PL, 2PL with lock conversions, Timestamp-ordering based protocol, Multi versioning protocol, Multi-granularity locking, Deadlock prevention protocols, Wait-die and wound-wait schemes, Time-out based schemes, Deadlock recovery, Nested transactions.		
MODULE 8:	**DATABASE RECOVERY TECHNIQUES**	**3 Hours**
Recovery concepts, Deferred updates technique, Immediate update technique, Shadow paging.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	1	2	-	-	-	-	-	-	2	3	-	-
CO2	3	3	2	2	2	-	-	-	-	-	-	2	3	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	2	3	2	-
CO4	3	2	2	2	3	-	-	-	-	-	-	2	3	3	-
CO5	3	2	2	3	-	-	-	1	-	-	-	2	3	3	-

CO6	3	3	3	2	-	-	-	-	-	-	-	2	3	3	-
	3	2.5	2.2	2	2.33			1				2	3	2.75	

Books:

1. Avi Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts,Tata McGraw - Hill Education.
2. Ramez Elmasri and Shamkant Navathe, Fundamentals of Database Systems, Publisher - Pearson Education, 5th Edition
3. Database systems, 6th edition, Ramez Elmasri, Shamkant B.Navathe, Pearson Education
4. Database Systems Design, Implementation, and Management, Peter Rob & Carlos Coronel, 7th Ed
5. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education
6. Microsoft SQL Server 2019 documentation: Databases - SQL Server | Microsoft Docs
7. Microsoft Azure SQL documentation: Azure SQL documentation - Azure SQL | Microsoft Docs
8. Microsoft Azure CosmosDB documentation: Introduction to Azure Cosmos DB |Microsoft Docs
9. Articles on Microsoft Azure and SQL Server: Sucharita Das, Author at SQLServerCentral

Operating System (TIU-UCS-T317)

Program: B. Tech. in CSE	**Year, Semester:** 3rd., 5th
Course Title: Operating Systems	**Subject Code:** TIU-UCS-T317
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Understand the structure, functions, and operations of operating systems, including computing environments and open-source OS.
2. Explore process management, scheduling, multithreading, and inter-process communication with synchronization techniques.
3. Learn memory management strategies, including virtual memory, paging, segmentation, and page replacement techniques.
4. Analyze deadlocks, including detection, prevention, and recovery, along with file system implementation and storage management.
5. Examine system protection, access control mechanisms, security policies, and cryptographic techniques for system security.

COURSE OUTCOME:

The students will be able to:

CO1:	Understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.	K2

CO2:	Analyze important algorithms e.g. Process scheduling and memory management algorithms, Disk scheduling algorithms.	K4
CO3:	Categorize the operating system's resource management techniques, dead lock management techniques, memory management techniques.	K4
CO4:	Demonstrate the ability to perform OS tasks in Red Hat Linux Enterprise.	K2
CO5:	Evaluate OS performance through scheduling, memory, and file system optimizations.	K4
CO6:	Develop shell scripts and system programs for process management and automation.	K3

COURSE CONTENT :

MODULE 1:	**INTRODUCTION TO OS**	**8 Hours**
Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems. System Structures: Operating System Services, User and Operating-System Interface, systems call, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.		
MODULE 2:	**PROCESS MANAGEMENT**	**10 Hours**
Process Concept: Process scheduling, Operations on processes, Inter-process communication, Communication in client server systems. Multithreaded Programming: Multithreading models, Thread libraries, Threading issues. Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling. Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Barriers, Classical IPC Problems – Dining philosophers problem, Readers and writers problem.		
MODULE 3:	**MEMORY MANAGEMENT**	**10 Hours**
Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation. Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.		
MODULE 4:	**DEADLOCKS & FILE SYSTEM**	**9 Hours**
Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention. File Systems: Files, Directories, File system implementation, management and optimization. Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.		
MODULE 5:	**SECURITY, SYSTEM PERFORMANCE**	**8 Hours**

System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights. System Security: Introduction, Program threats, System and network threats, Cryptography for security, User authentication, implementing security defenses, Firewalling to protect systems and networks, Computer security classification. Case Studies: Linux, Microsoft Windows.
TOTAL LECTURES | **45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	2	-	-	-	-	-	-	-	2	3	-	-
CO2	3	3	2	2	-	-	-	-	-	-	-	2	3	-	-
CO3	3	3	2	3	-	-	-	-	-	-	-	2	3	2	-
CO4	2	-	3	-	3	-	-	-	-	-	-	2	3	3	-
CO5	3	3	2	3	-	-	-	1	-	-	-	2	3	3	-
CO6	3	2	3	2	3	-	-	-	-	-	-	2	3	3	-
	2.83	2.6	2.4	2.4	3			1				2	3	2.75	

Books:

1. Operating System Concepts – Abraham Silberschatz, Peter B. Galvin, Greg Gagne
2. Modern Operating Systems – Andrew S. Tanenbaum, Herbert Bos
3. Operating Systems: Internals and Design Principles – William Stallings
4. Operating Systems: A Concept-Based Approach – Dhananjay M. Dhamdhere

Automata Theory (TIU-UCS-T323)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 5th
Course Title: Automata Theory	**Subject Code:** TIU-UCS-T323
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

1. To make the student aware about the basic concepts of different abstract computing methods.
2. To make the student aware about regular languages, regular grammar, regular expression, DFA, NFA, their relationship and closure properties of regular languages,
3. To make the students aware about context free languages(CFL), context free grammar, push down automata, closure properties of CFL, Chomsky normal form(CNF), Greibach normal Form(GNF)
4. To make the student aware about context sensitive grammar

COURSE OUTCOME:

After Completion of the course, the students will be able to:

CO1 :	Describe the concepts of formal theory of language, the meaning of computing and algorithms	K2
CO2 :	Describe and analyze different models of computing such as FA, CFG/PDA, TM	K4
CO3 :	Design above models for problem solving	K3
CO4 :	Analyze and identify the strengths and shortcomings of the above computing models	K4
CO5 :	Describe basic concepts of complexity theory: solvable and unsolvable problems, complexity classes, etc.	K2
CO6 :	Apply formal language and automata theory concepts to real-world computing problems.	K3

COURSE CONTENT:

MODULE 1:	**REGULAR LANGUAGES AND FINITE AUTOMATA**	**15 Hours**
Introduction, Alphabet, Language, and Grammar. Regular Expressions and Languages, Deterministic Finite Automata (DFA) and Equivalence with Regular Expressions, Nondeterministic Finite Automata (NFA) and Equivalence with DFA, Regular Grammars and Equivalence with Finite Automata, Properties of Regular Languages, Pumping Lemma for Regular Languages, Minimization of Finite Automata.		
MODULE 2:	**CONTEXT-FREE GRAMMAR/LANGUAGES**	**16 Hours**
Context-Free Grammars (CFG) and Context-Free Languages (CFL), Production, Parse Tree, and Derivation; Chomsky and Greibach Normal Forms, Non-deterministic Pushdown Automata (PDA) and Equivalence with CFG, Parse Trees, Ambiguity in CFG, Pumping Lemma for Context-Free Languages, Deterministic Pushdown Automata, Closure Properties of CFLs. Chomsky Hierarchy of Languages. Context-Sensitive Grammars: Context-Sensitive Grammars (CSG) and Context sensitive Languages (CSL), Linear Bounded Automata (LBA) and its Equivalence with CSG.		
MODULE 3:	**TURING MACHINES**	**9 Hours**
The Basic Model of Turing Machines (TM), Turing-Recognizable (Recursively Enumerable) and Turing-Decidable (Recursive) Languages and Their Closure Properties, Variants of Turing Machines, Non-deterministic TMs and its Equivalence with Deterministic TMs, Unrestricted Grammars and Equivalence with Turing Machines, TMs as Enumerators.		
MODULE 4:	**UNDECIDABILITY**	**5 Hours**
Church-Turing Thesis, Universal Turing Machine, The Universal and Diagonalization Languages, Reduction between Languages and Rice's Theorem, Undecidable Problems about Languages.		

TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	-	-
CO2	3	3	2	2	-	-	-	-	-	-	-	2	3	2	-
CO3	3	3	3	2	-	-	-	-	-	-	-	2	3	2	-
CO4	3	3	2	3	-	-	-	-	-	-	-	2	3	3	-
CO5	3	3	2	3	-	-	-	-	-	-	-	2	3	3	-
CO6	3	2	3	2	3	-	-	-	-	-	-	2	3	3	-
	3	2.66	2.4	2.4	3							2	3	2.6	

Books:

1. John E. Hopcroft, Rajeev Motwani , Jeffrey D. Ullman, Introduction to Automata Theory, Languages, And Computation, Pearson
2. Michael Sipser, Introduction to the Theory of Computation, Cengage
3. Dexter C. Kozen, Automata And Computability, Undergraduate Texts In Computer Science, Springer.
4. John Martin, Introduction To Languages And The Theory Of Computation, Tata Mcgraw Hill.

Harry R. Lewis And Christos H. Papadimitriou, Elements Of The Theory Of Computation, Pearson Education Asia.

Database Management System Lab (TIU-UCS-L315)

Program: B.Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Database Management System Lab	**Subject Code:** TIU-UCS-L315
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand the principles of relational databases and SQL.
2. Apply database operations using SQL to manage and manipulate data effectively.
3. Develop complex queries, stored procedures, and triggers for efficient data handling and automation.

COURSE OUTCOME:

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

CO-1	Understand and apply DDL (Data Definition Language) and DML (Data Manipulation Language) statements to create and modify database structures and data.	K3
CO-2	Perform join operations to retrieve data from multiple tables efficiently.	K3
CO-3	Use aggregate functions to analyze and summarize data within a database.	K4
CO-4	Ensure referential integrity and manage relationships between tables in a database.	K3
CO-5	Implement indexing and views to optimize database performance and retrieval operations.	K4
CO-6	Utilize transactions, stored procedures, and triggers to ensure data integrity and automate database operations.	K3

COURSE CONTENT:

MODULE 1:	**DDL AND DML OPERATIONS**	**9 Hours**
Introduction to SQL; DDL Statements: CREATE, ALTER, DROP; DML Statements: INSERT, UPDATE, DELETE; Constraints and data integrity		
MODULE 2:	**JOIN OPERATIONS**	**6 Hours**
Inner Join, Outer Join (Left, Right, Full); Cross Join, Self Join; Performance considerations in join operations		
MODULE 3:	**BUILT-IN FUNCTIONS AND INTEGRITY CONSTRAINTS**	**6 Hours**
Aggregate Functions (SUM, COUNT, AVG, MAX, MIN); String Functions; Integrity Constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY)		
MODULE 4:	**REFERENTIAL INTEGRITY**	**6 Hours**
Concept of Referential Integrity, Enforcing foreign key relationships, Handling cascading updates and deletes		
MODULE 5:	**INDEXING AND VIEWS**	**6 Hours**
Creating and using indexes, Advantages and limitations of indexes, Creating and managing views		
MODULE 6:	**STORED PROCEDURES, TRANSACTIONS, AND TRIGGERS**	**12 Hours**
Creating and executing stored procedures, Transaction Control (COMMIT, ROLLBACK, SAVEPOINT), Creating and managing triggers		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	1	-	1	3	-	-
CO2	2	3	-	-	3	-	-	-	-	2	-	-	3	-	-
CO3	2	3	-	-	3	-	-	-	-	2	-	-	3	1	-
CO4	3	2	-	-	3	-	-	-	-	1	-	-	3	-	1
CO5	2	-	3	-	3	-	-	-	-	-	2	-	3	2	2
CO6	2	-	3	-	3	-	-	-	-	-	2	-	3	2	3
	2.33	2.5	3		2.833					1.5	2	1	3	1.66	2

Books:

1. Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th ed.). Pearson.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts (7th ed.). McGraw-Hill.
3. Ramakrishnan, R., & Gehrke, J. (2014). Database Management Systems (3rd ed.). McGraw-Hill.
4. Date, C. J. (2019). An Introduction to Database Systems (8th ed.). Pearson.
5. Mullins, C. S. (2012). Database Administration: The Complete Guide to DBA Practices and Procedures (2nd ed.). Addison-Wesley.

Design and Analysis of Algorithms Lab (TIU-UCS-L321)

Program: B.Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Design and Analysis of Algorithms Lab	**Subject Code:** TIU-UCS-L321
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. To understand the fundamental concepts of algorithm design and analyze their time and space complexities.
2. To apply algorithmic techniques such as divide and conquer, dynamic programming, and greedy methods to solve computational problems.
3. To evaluate the efficiency and correctness of algorithms using mathematical analysis and empirical testing.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Develop and implement sorting algorithms such as Quick Sort and Merge Sort using the Divide and Conquer approach.	K3
CO-2	Apply dynamic programming techniques to solve optimization problems like the 0-1 Knapsack problem.	K3
CO-3	Employ algorithms such as Dijkstra's for solving single-source shortest path problems in graphs.	K4
CO-4	Analyze and examine algorithms like Floyd-Warshall's for finding the shortest path between all pairs of vertices in a graph.	K3
CO-5	Solve and optimize problems like the Travelling Salesman problem using various algorithmic approaches.	K4
CO-6	Evaluate the time and space complexity of algorithms using Big O notation and assess their performance in solving real-world problems.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO ALGORITHMS AND COMPLEXITY ANALYSIS**	**6 Hours**
Definition and Importance of Algorithms, Performance Analysis: Time and Space Complexity, Asymptotic Notation: Big-O, Big-Theta, and Big-Omega, Empirical and Theoretical Analysis: Experimental evaluation with iterative and recursive algorithms		
MODULE 2:	**SORTING AND SEARCHING ALGORITHMS**	**9 Hours**
Sorting Techniques: Bubble Sort, Insertion Sort, Selection Sort (basic sorting). Merge Sort: Divide and Conquer Strategy. Quick Sort: Partitioning and Randomized Pivot Selection. Heap Sort: Using Max-Heap and Min-Heap. Searching Techniques: Linear Search and Binary Search. Complexity Analysis: Best-case, Worst-case, and Average-case performance comparison.		
MODULE 3:	**GREEDY ALGORITHMS**	**6 Hours**
Greedy Methodology: Characteristics and Applicability. Applications: Fractional Knapsack Problem, Minimum Spanning Tree (MST) using Prim's and Kruskal's Algorithms. Complexity Analysis: Time and Space Complexity of Greedy Algorithms.		
MODULE 4:	**DYNAMIC PROGRAMMING (DP)**	**9 Hours**
Dynamic Programming Paradigm: Optimal Substructure and Overlapping Subproblems. Applications: 0/1 Knapsack Problem, Longest Common Subsequence (LCS), All-pairs shortest path using Floyd-Warshall Algorithm. Complexity Analysis: Time and Space Complexity of DP algorithms.		
MODULE 5:	**GRAPH ALGORITHM**	**6 Hours**
Graph Representation: Adjacency Matrix and Adjacency List. Graph Traversal: Breadth-First Search (BFS), Depth-First Search (DFS). Single-Source Shortest Path: Dijkstra's Algorithm.		
MODULE 6:	**BACKTRACKING AND BRANCH & BOUND**	**9 Hours**
Backtracking Concepts: Solving problems using backtracking. Applications: N-Queens Problem. Branch and Bound: Concept and Applications. Solving Traveling Salesman Problem (TSP). Complexity Analysis: Time complexity of backtracking and branch and bound.		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	1	-	-	-	1	-	2	2	-	
CO2	2	3	2	-	3	-	-	-	-	-	1	2	2	-	
CO3	2	3	3	1	2	-	-	-	-	-	-	2	3	-	
CO4	1	2	3	2	3	-	-	-	-	-	1	2	2	1	
CO5	2	2	3	1	2	-	-	-	-	-	-	3	3	-	
CO6	2	3	2	1	3	-	-	-	-	1	-	3	2	1	
	2	2.5	2.6	1.25	2.5	1				1	1	2.33	2.33	1	

Books:

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MITPress.
2. J. Kleinberg and E. Tardos,AlgorithmDesign,Addison-Wesley.
3. Harry R. Lewis and Larry Denenberg, Data Structures and their Algorithms, Harper Collins.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press.
5. E. Horowitz and S. Sahani, Fundamentals of Computer Algorithms, Computer Science Press.

Object-Oriented Systems Lab (TIU-UCS-L319)

Program: B.Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Object-Oriented Systems Lab	**Subject Code:** TIU-UCS-L319
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. obtain the foundational skills to write, compile, and execute basic Java programs, while exploring the use of data types, variables, arrays, and control structures (decision-making and loop control).
2. implement object-oriented principles such as data abstraction, encapsulation, polymorphism, inheritance, interfaces, and packages, using Java, to solve real-world programming problems effectively.
3. create robust Java programs utilizing exception handling, multi-threading, and applet programming, while focusing on debugging, evaluating program correctness, and ensuring code efficiency and maintainability.

COURSE OUTCOME:

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

CO-1	Demonstrate the ability to write, compile, and execute basic Java programs.	K3
CO-2	Illustrate the use of data types, variables, arrays, and control structures such as decision-making (if, nested if) and loop control (do, while, for).	K3

CO-3	Apply concepts of data abstraction, encapsulation, polymorphism, inheritance, interfaces, and packages to solve problems in Java.	K4
CO-4	Develop Java programs incorporating exception handling and multi-threading mechanisms.	K3
CO-5	Execute applet programs and illustrate their usage.	K4
CO-6	Evaluate and debug Java programs for correctness, performance, and maintainability, ensuring efficient use of resources and adherence to best coding practices.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO JAVA PROGRAMMING BASICS**	**9 Hours**
Overview of Java programming language, IDE setup, and compiling Java programs; Understanding the basic structure of a Java program, including classes, methods, and variables; Overview of primitive data types (int, float, char, etc.); Operators: Arithmetic, relational, logical, and assignment operators; Introduction to decision-making statements in Java (if, nested if, switch).		
MODULE 2:	**LOOP CONTROL STRUCTURES AND ARRAYS**	**9 Hours**
Implementing for, while, and do-while loops for repeating code execution; Nested loops for multi-level iteration; string operations such as substring(), length(), charAt(), etc.; creation and manipulation of single-dimensional and multi-dimensional arrays.		
MODULE 3:	**CLASSES, OBJECTS, AND STATIC MEMBERS**	**6 Hours**
Understanding the basic concepts of classes and objects in Java; Using constructors, instance methods, and instance variables; concept of static members; significance of static variables, methods, and static blocks in Java.		
MODULE 4:	**OBJECT-ORIENTED CONCEPTS: INHERITANCE, POLYMORPHISM, AND ABSTRACTION.**	**6 Hours**
Concept of inheritance in Java: extending classes, constructor chaining, method overriding; Types of inheritance: single, multilevel, and hierarchical inheritance; Understanding the use of abstract classes and abstract methods; Use cases for abstract classes in Java.		
MODULE 5:	**PACKAGES, EXCEPTION HANDLING, AND MULTITHREADING**	**6 Hours**
Introduction to Java packages and their role in organizing code; Demonstrating the use of built-in packages (e.g., java.util); Basics of exception handling: try, catch, throw, throws, and finally; Creating custom exceptions and handling multiple exceptions.		
MODULE 6:	**APPLET PROGRAMMING AND GUI DEVELOPMENT WITH AWT**	**9 Hours**
Introduction to applet programming: lifecycle methods (init(), start(), stop(), destroy()); Differences between applets and applications; Overview of GUI programming in Java using AWT; Working with basic GUI components: Button, Label, TextField, etc.; Event handling in AWT components.		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO1	3	2	3	-	2	-	-	-	-	1	-	2	3	1	2
CO2	3	2	3	-	3	-	-	-	-	1	-	2	3	1	3
CO3	3	3	3	2	3	-	-	-	1	2	-	2	3	2	3
CO4	3	2	3	2	3	-	-	-	1	2	-	2	3	2	3
CO5	2	1	2	-	3	-	-	-	-	1	-	3	2	1	2
CO6	3	3	3	2	3	-	-	-	1	2	-	3	3	2	3
	2.833	2.16	2.83	2	2.833				1	1.5		2.33	2.8333	1.5	2.666

Books:

1. "Java: The Complete Reference" by Herbert Schildt.
2. "Core Java Volume I—Fundamentals" by Cay S. Horstmann.
3. "Head First Java" by Kathy Sierra and Bert Bates.
4. "Effective Java" by Joshua Bloch.

Operating System Lab (TIU-UCS-L317)

Program: B.Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Operating System Lab	**Subject Code:** TIU-UCS-L317
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Understand fundamental operating system concepts such as processes, threads, memory management, and inter-process communication (IPC).
2. Implement and analyze core OS functionalities, including scheduling, file management, and synchronization.
3. Gain hands-on experience with Red Hat Enterprise Linux and practical troubleshooting of OS-related issues.

COURSE OUTCOME:

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

CO-1	Explain fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.	K3
CO-2	Analyze important algorithms such as process scheduling, memory management, and disk scheduling algorithms.	K4
CO-3	Categorize the operating system's resource management techniques, deadlock management techniques, and memory management techniques.	K4
CO-4	Demonstrate the ability to perform OS tasks in Red Hat Linux Enterprise.	K3
CO-5	Implement and test operating system concepts like process synchronization,	K4

	inter-process communication (IPC), and file management in a practical environment.	
CO-6	Evaluate and troubleshoot operating system performance, addressing resource allocation, process management, and system stability issues.	K4

COURSE CONTENT:

MODULE 1:	**PROCESS MANAGEMENT AND SCHEDULING**	**9 Hours**
Concept of processes and threads, CPU scheduling algorithms (FCFS, SJF, RR, Priority), Process creation and management in Linux.		
MODULE 2:	**INTER-PROCESS COMMUNICATION AND SYNCHRONIZATION**	**9 Hours**
IPC mechanisms (pipes, message queues, shared memory), Process synchronization, Semaphores, and mutex locks.		
MODULE 3:	**MEMORY MANAGEMENT TECHNIQUES**	**6 Hours**
Paging and segmentation, Virtual memory, Page replacement algorithms (FIFO, LRU, Optimal).		
MODULE 4:	**FILE SYSTEM AND DISK MANAGEMENT**	**6 Hours**
File operations, File allocation methods, Disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN).		
MODULE 5:	**DEADLOCK HANDLING AND RESOURCE ALLOCATION**	**6 Hours**
Deadlock prevention and avoidance, Banker's algorithm, Resource allocation graphs.		
MODULE 6:	**SYSTEM PERFORMANCE AND SECURITY**	**9 Hours**
Monitoring system performance, Troubleshooting OS issues, and Security management in Linux.		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO2	3	2	-	-	3	-	-	-	-	-	-	-	3	-	-
CO3	3	3	2	2	-	-	-	-	1	-	-	-	3	-	-
CO4	3	2	-	2	3	-	-	-	1	-	-	-	3	-	-
CO5	2	1	2	-	3	-	-	-	-	-	-	2	2	-	-
CO6	3	2	-	2	3	-	-	-	1	-	-	2	3	-	-
	2.8333	2	2	2	2.8				1			2	2.833		

Books:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. - Operating System Concepts (10th ed.), Wiley
2. Tanenbaum, A. S., & Bos, H. - Modern Operating Systems (4th ed.), Pearson
3. Dhamdhere, D. M. - Operating Systems: A Concept-Based Approach (3rd ed.), McGraw-Hill
4. Mauro, J., & McDougall, R. - Solaris Internals: Core Kernel Architecture, Prentice Hall

Mobile Computing (TIU-UCS-S301A)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Mobile Computing	**Subject Code:** TIU-UCS-S301A
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

Enable the student to:

1. To provide foundational knowledge of mobile communication systems by exploring cellular networks, multiple access technologies, and the evolution from 1G to 5G.
2. To analyze network and transport layer challenges in mobile environments by studying wireless standards, Mobile IP, TCP variants, and mobile routing protocols.

To introduce advanced topics in mobile computing such as Wireless Sensor Networks, Cognitive Radio, Internet of Things (IoT), and Software-Defined Networking (SDN) to prepare students for research and industry applications

COURSE OUTCOME:

After Completion of the course, the students will be able to:

CO-1	Explain the fundamentals of mobile communication, cellular networks, and multiple access technologies.	K2
CO-2	Analyze wireless network standards, Mobile IP, and transport layer protocols for mobile environments.	K4
CO-3	Compare mobile routing algorithms and multicast protocols in MANETs.	K4
CO-4	Understand Wireless Application Protocol (WAP) and its role in mobile internet access.	K2
CO-5	Explore advanced mobile network technologies like 5G, MIMO, cognitive radio, and wireless sensor networks.	K3
CO-6	Apply emerging concepts like Software-Defined Networking (SDN) and IoT in mobile computing.	K3

COURSE CONTENT:

MODULE 1:	**Introduction to Mobile Communication**	15 hours

Introduction to mobile wireless communication and systems, Description of cellular system, Channel interferences. Channel assignment schemes. Concept of 1G, Multiple Access Technologies in cellular communication: Time division multiple access (TDMA), Frequency division multiple access (FDMA), Code Division Multiple Access (CDMA). Second generation (2G) Network: Global system for mobile communication (GSM). 2.5G Wireless Networks-GPRS, CDMA (IS 95), Third Generation 3G Wireless Networks-UMTS, Fourth Generation 4G Wireless Networks-LTE Advanced.		
MODULE 2:	**Mobile Network and Transport Layer**	10 hours
Wireless LAN–IEEE 802.11, PAN-Bluetooth- Piconet, Scatternet, Connection Establishment, Protocol Stack, Recap of Mobile IP, Traditional TCP, Indirect TCP, Snooping TCP, Mobile TCP, ATCP, Transmission / Timeout Freezing Selective Retransmission, Transaction oriented TCP.		
MODULE 3:	**Mobile Routing and Application Protocols**	10 hours
Mobile Ad hoc Networks (MANETs): Overview, Properties of a MANET, routing and various routing algorithms- DSR, WRP, DSDV, AODV,ZRP. Multicast Routing Algorithms: MAODV, ODMRP. Wireless Application Protocol (WAP): The Mobile Internet standard, WAP Gateway and Protocols, wireless markup Languages (WML).		
MODULE 4:	**Advanced Issues in Mobile Network**	10 hours
Wireless Sensor Network, Fifth Generation (5G) Wireless Networks: MIMO System Design and Channel Allocation schemes; Convex Optimization based treatment, Cognitive Radio and Internet of Things. SDN.		
Total		**45 hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	0	1	1	0	1	0	1	0	2	3	2	2
CO2	3	3	2	1	2	0	0	1	1	1	0	2	3	2	2
CO3	2	2	2	1	3	0	0	0	1	1	1	2	3	2	3
CO4	2	2	2	0	2	1	1	1	0	1	0	2	2	2	3
CO5	3	2	3	1	3	1	1	1	0	1	1	3	3	3	3
CO6	3	2	3	1	3	1	1	1	1	1	1	3	3	3	3
	2.67	2.17	2.17	0.67	2.3	0.7	0.5	0.83	0.5	1	0.5	2.33	2.83	2.33	2.67

Recommended Books:

Main Reading

1. Wireless Networks: Applications and Protocols, T.S. Rappaport, Pearson Education

2. Wireless Communications, A. Goldsmith, Cambridge University Press.
3. Wireless Communication: Stallings, Pearson.

Supplementary Reading

1. Mobile Communications, Jochen Schiller, 2nd Edition, Pearson Education, India.
2. NPTEL Materials from the course of Convex Optimization offered by Aditya P. Jagannatham.
3. Prototyping and Load Balancing the Service Based Architecture of 5G Core using NFV by Vamshi Kiran Buyakar, Harsh Agarwal, Bheemarjuna Reddy Tamma, and Antony Franklin (Indian Institute of Technology Hyderabad), published in IEEE NETSOFT 2019.

Cryptography & Network Security (TIU-UCS-S301B)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Cryptography & Network Security	**Subject Code:** TIU-UCS-S301B
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

1. To make the student understand the core principles of cryptography.
2. To develop mathematical proficiency in cryptographic system.
3. To make the student aware of cryptographic protocols and their applications.

COURSE OUTCOME:

The students will be able to:

CO-1:	Familiarization with Cryptographic Terminologies.	K2
CO-2:	Understanding of mathematical tools for cryptography	K3
CO-3:	Comprehension of Symmetric and Asymmetric Cryptographic Techniques	K2
CO-4:	Application of Public-Key Cryptosystems and Secure Key Exchange Methods	K3
CO-5:	Analysis of Hash Functions and Digital Signature Mechanisms.	K4
CO-6:	Exploration of Advanced Cryptographic Methods and Security Standards	K4

COURSE CONTENT:

MODULE 1	Introduction	4 Hours
Basic objectives of cryptography, secret-key and public-key cryptography, one-way and trapdoor one-way functions, cryptanalysis, attack models, classical cryptography.		
MODULE 2	**Block and Stream Ciphers**	**13 Hours**
Block ciphers: Modes of operation, DES and its variants, RCS, IDEA, SAFER, FEAL, BlowFish, AES, linear and differential cryptanalysis. **Stream ciphers:** Stream ciphers based on linear feedback shift registers, SEAL, unconditional security.		
MODULE 3	**Public-key Cryptosystem**	**10 Hours**
Public-key parameters: Modular arithmetic, gcd, primality testing, Chinese remainder theorem and its application in modular inverse, modular square roots, finite fields. RSA, Rabin and EIGamal schemes,		
MODULE 4	**Key exchange:**	**2 Hours**
Diffie-Hellman and MQV		
MODULE 5	**Message digest**	**3 Hours**
Properties of hash functions, MD2, MD5 and SHA-1, keyed hash functions, attacks on hash functions		
MODULE 6	**Digital signatures:**	**2 Hours**
RSA, DSA and NR signature schemes, blind and undeniable signatures.		
MODULE 7	**Intractable problems**	**2 Hours**
Integer factorization problem, RSA problem, modular square root problem, discrete logarithm problem, Diffie-Hellman problem, known algorithms for solving the intractable problems.		
MODULE 8	**Entity authentication and Network security, Standards**	**4 Hours**
Passwords, challenge-response algorithms, zero-knowledge protocols. Network security: Certification, public-key infra-structure (PKI), secure socket layer (SSL), Kerberos Standards: IEEE, RSA and ISO standards.		
MODULE 9	**Advanced topics**	**5 Hours**
Elliptic and hyper-elliptic curve cryptography, number field sieve, lattices and their applications in cryptography, hidden monomial cryptosystems, cryptographically secure random number generators.		
TOTAL		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	1	1	1	0	0	1	0	1	0	2	2	2	1
CO2	3	3	1	2	2	0	0	0	0	1	0	2	3	2	2
CO3	2	3	2	2	2	1	0	1	0	1	0	2	3	3	2
CO4	2	2	2	2	3	1	0	1	0	1	0	2	3	3	3
CO5	3	3	2	2	2	1	0	2	1	2	1	2	3	3	3
CO6	3	3	2	2	3	2	1	2	1	2	1	3	3	3	3
	2.5	2.7	1.7	1.8	2.2	0.8	0.2	1.2	0.3	1.3	0.3	2.2	2.8	2.7	2.3

Recommended Books:

Main Reading

1. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press.
2. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall of India.
3. Neal Koblitz, A course in number theory and cryptography, Springer.

Supplementary Reading

1. Johannes A. Buchmann, Introduction to Cryptography, Undergraduate Text in Mathematics, Springer.
2. 2. Doug Stinson, Cryptography Theory and Practice, CRC Press.
3. 3. Das and C. E. VeniMadhavan, Public-Key Cryptography: Theory and Practice, Pearson Education Asia.

Information and Coding Theory (TIU-UCS- S301E)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Information and Coding Theory	**Subject Code:** TIU-UCS-S301E
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

1. To introduce students to the principles of information theory, entropy.
2. To make students aware of error detecting and error correcting codes.
3. To make students enable to analyse, implement encoding and decoding techniques.
4. To make students able so that they can apply coding techniques in real world applications, such as data storage systems

COURSE OUTCOME:

The students will be able to:

CO-1:	Explain fundamental concepts of information theory, entropy, and coding.	k2
CO-2:	Apply source coding techniques (Shannon-Fano, Huffman, Lempel-Ziv) for data compression.	K3
CO-3:	Analyze linear block codes and cyclic codes for error detection and correction.	K3
CO-4:	Implement convolutional codes, state diagrams, trellis structures, and decoding algorithms.	K3

CO-5:	Evaluate the performance of sequential and maximum likelihood decoding algorithms.	K4
CO-6:	Apply BCH codes and Galois field computations for real-world error control applications.	K3

COURSE CONTENT:

MODULE 1	**Introduction**	**2Hours**
Information Theory: Uncertainty and information, average mutual information and entropy.		
MODULE 2	**Source Coding**	**5Hours**
: Mathematical model of Information, A Logarithmic Measure of Information, Average and Mutual Information and Entropy, coding for Discrete less sources, Source coding theorem, fixed length and variable length coding, properties of prefix codes.		
MODULE 3	**Data Compression and Source Coding Techniques**	**4Hours**
Shannon-Fano Coding, Huffman code, Huffman code applied for pair of symbols, efficiency calculations, Lempel-Ziv codes		
MODULE 4	**Linear Block Codes**	**5Hours**
Introduction to Linear block codes, Generator Matrix, Systematic Linear Block codes, Encoder Implementation of Linear Block Codes, Parity Check Matrix, Syndrome testing, Error Detecting and correcting capability of Linear Block codes.		
MODULE 5	**Error Detection and Correction using Block Codes**	**4Hours**
Hamming Codes, Probability of an undetected error for linear codes over a Binary Symmetric Channel, Weight Enumerators and Mac-Williams identities, Perfect codes, Application of Block codes for error control in data storage Systems		
MODULE 6	**CyclicCodes**	**4Hours**
Algebraic structure of cyclic codes, Binary Cyclic code properties, Encoding in systematic and non-systematic form, Encoder using (n-k) bit shift register, Syndrome Computation and Error detection, Decoding of Cyclic Codes.		
MODULE 7	**Convolutional Codes**	**7 Hours**
Encoding of Convolutional codes, Structural properties of Convolutional codes, state diagram, Tree diagram, Trellis Diagram, maximum, Likelihood decoding of Convolutional codes.		
MODULE 8	**Decoding Algorithms for Error Correction**	**4 Hours**
Viterbi Algorithm, Fano, Stack Sequential decoding algorithms, Application of Viterbi and sequential decoding.		
MODULE 9	**Bch Codes**	**10 Hours**

Groups, fields, binary Fields arithmetic, construction of Falois fields GF (2m), Basic properties of Falois Fields, Computation using Falois Field GF (2m) arithmetic, Description of BCH codes, Decoding procedure for BCH codes.	
Total	45 hours

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	2	2	1	1	1	2	2	1	2	3	3	2
CO2	3	2	3	2	3	2	1	1	2	2	1	2	3	3	3
CO3	3	3	2	3	3	2	1	1	3	3	1	2	2	3	3
CO4	3	3	3	3	3	2	1	1	3	3	1	2	3	3	3
CO5	2	3	3	3	3	2	1	1	3	3	1	2	3	3	3
CO6	3	3	3	3	3	2	1	1	3	3	1	2	3	3	3
	2.83	2.66	2.5	2.66	2.8	1.8	1	1	2.66	2.66	1	2	2.83	3	2.83

Recommended Books:

Main Reading

1. T. M. Cover, J. A. Thomas, Elements of Information Theory, Wiley
2. R. Togneri, C.J.S deSilva, Fundamentals of Information Theory and Coding Design, Taylor and Francis
3. Principles of Digital Communication – Das, Mukherjee, Chatterjee
4. Coding and Information Theory by Richard W. Hamming
5. Handbook of Coding Theory, Vol 1 & 2, by V. S. Pless and W. C. Huffman

Supplementary Reading

1. R. J. McEliece, The Theory of Information and Coding, Cambridge University Press
2. R. Bose, Information Theory Coding and Cryptography, Tata McGraw Hill
3. Introduction to Error Control Codes – S. Gravano
4. Error Control Coding: Fundamentals and Applications – Shu Lin, Danilel J. Costello, Jr.
5. The Theory of Error-Correcting Codes, Vol 1 & 2, by F.J. MacWilliams and N.J.A. Sloane
6. Algebraic Codes for Data Transmission by Richard E. Blahut
7. Introduction to Coding Theory by Jacobus Hendricus van Lint
8. Coding and Information Theory by Steven Roman
9. Error Control Coding by Shu Lin and Daniel J. Costello
10. Error Correction Coding: Mathematical Methods and Algorithms by Todd K. Moon

Artificial Intelligence (TIU-UCS-S301C)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 5th.
Course Title: Artificial Intelligence	**Subject Code:** TIU-UCS-S301C

Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVES:

1. Understand the core concepts, history and evolutions of Artificial Intelligence
2. Explore search and optimization techniques like heuristic and uninformed search, evolutionary algorithms
3. Develop Logical and Probabilistic Reasoning like Bayesian network, knowledge representations
4. Apply Machine Learning Concepts like supervised, unsupervised, reinforcement learning

COURSE OUTCOMES:

The students will be able to

CO1:	Understand the fundamental concepts Artificial Intelligence such as knowledge representation, problem solving and expert systems	K2
CO2:	Understand the use of AI to solve communication problems using Natural Language Processing	K2
CO3:	Develop knowledge of decision making and learning methods.	K3
CO4:	develop new facts from existing knowledge base using resolution and unification.	K4
CO5:	Demonstrate the way of writing Facts and Rules to solve some problems based on rules and to develop systems for question-answer.	K4
CO6:	Apply AI techniques like heuristic search, genetic algorithms, and neural networks to solve real-world problems.	K3

Course Content

MODULE 1:	**Basics of AI**	**8 Hours**
Introduction: Overview of Artificial intelligence- Problems of AI, AI technique, Tic - Tac - Toe problem., Intelligent Agents: Agents & environment, nature of environment, structure of agents, goal-based agents, utility-based agents, learning agents., Learning: Forms of learning, inductive learning, learning decision trees, explanation-based learning, learning using relevant information, neural net learning & genetic learning.		
MODULE 2:	**Different types of searching algorithms, Problem Solving**	**13 Hours**
Problems, Problem Space & search: Defining the problem as state space search, production system, constraint satisfaction problems, issues in the design of search programs, Search techniques: Solving problems by searching: Problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search, depth limited search, bidirectional search, comparing uniform search strategies, Heuristic search strategies: Greedy best-first search, A* search, memory bounded heuristic search: local search algorithms & optimization problems: Hill climbing search, simulated annealing search, local beam search, genetic algorithms; constraint satisfaction problems, local search for constraint satisfaction problems, Adversarial search: Games, optimal decisions & strategies in games, the minimax search procedure, alpha-beta pruning, additional refinements, iterative deepening.		
MODULE 3:	**Knowledge & Reasoning, Knowledge & Reasoning**	**12 Hours**
Knowledge representation issues, representation & mapping, approaches to knowledge representation, issues in knowledge representation, Using predicate logic: Representing simple fact in logic, representing instant & ISA relationship, computable functions & predicates, resolution, natural deduction, Representing knowledge using rules: Procedural vs declarative knowledge, logic		

programming, forward vs backward reasoning, matching, control knowledge, Probabilistic reasoning: Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Fuzzy sets, and fuzzy logics, belief propagation. Markov processes, and Hidden Markov models.		
MODULE 4:	**Different fields of AI, Natural Language Processing**	**12 Hours**
Introduction, Syntactic processing, semantic analysis, discourse, and pragmatic processing., Expert Systems: Representing and using domain knowledge, expert system shells, and knowledge acquisition. Basic knowledge of programming languages like Prolog.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	1	-	1	3	3	2
CO2	2	2	-	-	-	-	-	-	-	-	-	1	2	3	3
CO3	2	3	-	-	1	-	-	-	-	-	-	1	2	3	3
CO4	2	2	-	-	-	-	-	-	-	-	-	-	2	2	2
CO5	2	2	-	-	1	-	-	-	-	1	-	-	2	2	2
CO6	2	2	3	2	1	-	-	-	-	-	1	2	3	3	3
	2.17	2.17	3	2	1					1	1	1.25	2.33	2.67	2.5

Recommended Books:

Main Reading

1. Artificial Intelligent e: Elaine Rich, Kevin Knight, Mc-Graw Hill.
2. Introduction to AI & Expert System: Dan W. Patterson, PHI.
3. Artificial Intelligent by Luger (Pearson Education)
4. Russel & Norvig, Artificial Intelligent e: A Modern Approach, Pearson Education

SAP (TIU-UCS-S301F)

Program: B. Tech. in CSE	**Year, Semester: 3rd**, 5th
Course Title: SAP	**Subject Code:** TIU-UCS-S301F
Contact Hours/Week: 2–0–0 (L–T–P)	**Credit:** 2

COURSE OBJECTIVE:

Enable the student to:

1. understand SAP architecture and ABAP programming concepts, including program flow, object navigator, and transaction management.

2. develop proficiency in ABAP programming, covering statements, loops, modularization, OOP, and database handling.
3. enhance data modeling and reporting skills using transparent tables, Open SQL, and ALV grid control.

COURSE OUTCOME:

The students will be able to:

CO-1	Understand SAP System Architecture and program flow.	K2
CO-2	Develop ABAP programs using SAP Object Navigator and Repository.	K3
CO-3	Implement ABAP statements, logical expressions, and loops.	K3
CO-4	Design and manage ABAP structures and transparent tables.	K4
CO-5	Apply object-oriented programming concepts in ABAP.	K3
CO-6	Utilize Open SQL, database handling, and ALV reporting in SAP.	K4

COURSE CONTENT:

MODULE 1:	**SAP SYSTEM ARCHITECTURE & ABAP BASICS**	**10 Hours**
SAP System Architecture, Flow of a Program, SAP Object Navigator, Repository, Creating Packages, Developing ABAP Programs, Creating Transactions, Adding Transactions to Favorites		
MODULE 2:	**ABAP PROGRAMMING FUNDAMENTALS**	**10 Hours**
Basic ABAP Statements, ABAP Structures, Logical Expressions, Conditional Statements, Loops, Search Helps (F4), String Manipulation, Selection Screens (Radio Button, Check Box)		
MODULE 3:	**ABAP OBJECT-ORIENTED PROGRAMMING & DATA HANDLING**	**10 Hours**
Object-Oriented Programming (Classes, Objects, Methods, Interfaces), Creating Structures in ABAP, Transparent Tables (Data Modeling, Table Creation, Maintenance, Viewing Data), Database Handling (Open SQL, Modifications, Data Retrieval, SQL JOINs)		
MODULE 4:	**ADVANCED ABAP PROGRAMMING**	**10 Hours**
ABAP Subroutines (Procedures, Modularization, Include Programs), ALV Grid Control (ALV Programming, ALV Report Generation)		
MODULE 5:	**SAP APPLICATION DEVELOPMENT & BEST PRACTICES**	**5 Hours**
Best Practices in ABAP Development, Debugging Techniques, Performance Optimization, Real-world Use Cases, Project-based Learning & Hands-on Practice		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	-	1	3	-	-
CO2	3	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	-	3	-	-

CO4	3	2	-	2	3	-	-	-	-	-	-	1	3	-	-
CO5	2	1	2	-	3	-	-	-	-	-	-	1	2	-	-
CO6	3	2	-	2	3	-	-	-	-	-	-	2	3	-	-
	2.833	2	2	2	2.6							1.25	2.833		

Books:

1. Berg, B. O., & Moxon, P. (2009). SAP ABAP Programming for Beginners. SAP Press.
2. Haas, S., & Mathew, B. (2019). ABAP Development for SAP S/4HANA. SAP Press.
3. Hardy, P. (2021). ABAP to the Future. SAP Press.
4. Haeuptle, K. (2020). Clean ABAP: A Style Guide for Developers. SAP Press.
5. Keller, H. (2009). ABAP Programming Guidelines. SAP Press.
6. Keller, H., & Krüger, S. (2007). SAP ABAP Objects. SAP Press.
7. Kogent Learning Solutions Inc. (2011). SAP ABAP Handbook. Tata McGraw-Hill Education.
8. Lloyd, K. (2012). SAP ABAP: Advanced Cookbook. Packt Publishing.
9. McGhee, D. (2014). ALV Reports in SAP. SAP Press.
10. Wood, J. (2015). Object-Oriented Programming with ABAP Objects. SAP Press.

Generative AI (TIU-UCS-S301D)

Program: B. Tech. in CSE	Year, Semester: 3rd Yr., 5th Sem.
Course Title: Generative AI	Subject Code:TIU-UCS-S301D
Contact Hours/Week: 3–0–0 (L–T–P)	Credit: 3

COURSE OBJECTIVE:

Enable the student to:

1. To introduce the fundamental concepts and techniques of Generative AI.
2. To explore different generative models, including VAEs, GANs, Transformers, and Diffusion Models.
3. To analyze applications, challenges, and ethical concerns in Generative AI.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	Understand the fundamental concepts of Generative AI, including different generative models and their applications.	K2
CO-2:	Explain the working principles of Variational Autoencoders (VAEs) and their role in probabilistic modeling.	K2

CO-3:	Analyze the architecture and training challenges of Generative Adversarial Networks (GANs) for synthetic data generation.	K4
CO-4:	Evaluate transformer-based generative models like GPT and their applications in natural language generation.	K5
CO-5:	Compare different generative techniques such as Diffusion Models, Neural Style Transfer, and AI-driven multimedia synthesis.	K4
CO-6:	Assess the ethical, societal, and future implications of Generative AI, including bias, deepfakes, and responsible AI practices.	K5

COURSE CONTENT:

MODULE 1:	Introduction to Generative AI	5 Hours
Introduction to AI, Machine Learning, and Deep Learning, Generative vs. Discriminative Models, Overview of Generative AI Applications (Image Generation, Text Synthesis, Audio Synthesis), Introduction to Probability Distributions in Generative AI, Basics of Deep Generative Models		
MODULE 2:	Variational Auto encoders (VAEs)	7 Hours
Basics of Autoencoders and their Limitations, Introduction to Variational Autoencoders (VAEs), Encoder-Decoder Architecture and KL Divergence, Training and Optimization of VAEs, Applications of VAEs in Image and Text Generation.		
MODULE 3:	Generative Adversarial Networks (GANs)	7 Hours
Fundamentals of GANs: Generator and Discriminator, Training Process and Loss Functions, Challenges in Training GANs (Mode Collapse, Instability), Types of GANs: DCGAN, CGAN, WGAN, StyleGAN, Applications of GANs in Image Synthesis, Data Augmentation, and Deepfake Generation		
MODULE 4:	Transformer-Based Generative Models	8 Hours
Introduction to Transformers and Self-Attention Mechanism, BERT vs. GPT: Understanding Differences, GPT Architecture and Training Methodology, Large Language Models (LLMs) and Their Applications in Text Generation, Ethical Considerations and Limitations of Transformer Models		

MODULE 5:	Diffusion Models & Other Generative Techniques	8 Hours
Introduction to Diffusion Models for Image Generation, Working Principle of Stable Diffusion and DALL·E, Score-Based Generative Models, Neural Style Transfer and Image-to-Image Translation AI-Generated Audio, Speech Synthesis, and Music Composition		
MODULE 6:	Applications, Challenges, and Future of Generative AI	10 Hours
Generative AI in Healthcare, Art, and Scientific Research, Legal and Ethical Concerns: Deepfakes, Bias, and Copyright Issues, Responsible AI and Guidelines for Safe Generative AI Usage Future Trends: Multimodal AI, Real-Time Generation, and Personalization, Research Challenges and Open Problems in Generative AI		
TOTAL LECTURES		45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	-	1	3	-	-
CO2	3	2	-	-	2	-	-	-	-	-	-	-	3	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	-	3	-	-
CO4	3	2	-	2	3	-	-	-	-	-	-	1	3	-	-
CO5	2	1	2	-	3	-	-	-	-	-	-	1	2	-	-
CO6	3	2	-	2	3	-	-	-	-	-	-	2	3	-	-
	2.83	2	2	2	2.6							1.25	2.83		

Books:

1. BKP Horn , Robot Vision, Mit Press

Dana Harry Ballard, Christopher M. Brown, Computer vision, Prentice-Hall.

SEMESTER 6

Operations Research & Optimization Techniques (TIU-UMA-T302)

Program: B. Tech. in CSE	**Year, Semester:** 3rd., 6th
Course Title: Operations Research & Optimization Techniques	**Subject Code:** TIU-UMA-T302
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. understand the importance and value of Operations Research in real life, and finding solutions to various real-life problems
2. formulate linear programming problem from verbal description, and finding solutions
3. learn the basics in the field of game theory and solution techniques of various problems
4. choose the appropriate queuing model for a given practical application and finding solutions
5. draw a network diagram and determine related time, path, etc.

COURSE OUTCOME:

On completion of the course, the student will be able to:

CO-1:	analyze any real-life system with limited constraints, present it in a linear programming form and hence find its solution.	K4
CO-2:	solve both balanced and unbalanced transportation problems using various methods.	K3
CO-3:	determine solutions of a variety of problems mathematically such as assignment, travelling salesman etc. and associate how real-life problems are depicted.	K4
CO-4:	formulate game models and solve them by utilizing different methods.	K4

CO-5:	choose the appropriate queuing situations and deduce the optimal solutions using models for different situations.	K3
CO-6:	construct network diagrams for service and manufacturing systems, and find related time, path, etc.	K4

COURSE CONTENT:

MODULE 1:	**Linear Programming**	**14 Hours**
Formulation of Linear Programming Problem, Linear dependence and independence, Basic solutions, Convex Sets, Graphical Method, Simplex Method		
MODULE 2:	**Transportation and Assignment Problem**	**12 Hours**
Formulation of Transportation Problem, Initial Feasible Solution Methods, Optimality Test, Degeneracy in TP; Assignment Problem, Hungarian Method, Travelling Salesman Problem		
MODULE 3:	**Game Theory**	**8 Hours**
Two Person Zero Sum Game, Pure and Mixed Strategies, Algebraic Solution Procedure, Graphical Solution		
MODULE 4:	**Introduction to Queuing Models**	**5 Hours**
Elements of Queuing Model, Pure Birth Death Model.		
MODULE 5:	**Network Analysis**	**6 Hours**
CPM review, Crashing of an activity, Crash-cost slope, Time-cost trade		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	-	1	-	-	-	-	-	-	1	2	-	-
CO2	3	2	-	-	2	-	-	-	-	-	-	-	2	-	-
CO3	2	3	-	-	1	-	-	-	-	-	-	-	2	1	-
CO4	2	2	-	-	3	-	-	-	-	-	-	-	2	-	-
CO5	2	3	-	-	2	-	-	-	-	-	-	-	1	-	2
CO6	3	2	2	-	3	-	-	-	-	-	1	-	2	-	-
	2.5	2.5	2		2						1	1	1.833	1	2

Books:

1. Ghosh, M. K., & Chakraborty, S. (2010). Linear programming and game theory. Prentice-Hall of India.
2. Taha, H. A. (2017). Operations research: An introduction (10th ed.). Pearson.
3. Sharma, J. K. (2017). Operations research: Theory and applications (6th ed.). Macmillan Publishers India.
4. Sharma, S. D. (2009). Operations research. Kedar Nath Ram Nath.
5. Swarup, K., Gupta, P. K., & Man Mohan. (2014). Operations research. Sultan Chand & Sons.

Computer Network (TIU-UCS-T304)

Program: B. Tech. in CSE	**Year, Semester:** 3rd., 6th
Course Title: Computer Network	**Subject Code:** TIU-UCS-T304
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

4. understand network fundamentals including network types and topologies.
5. Analyze Network Protocols including TCP/IP, UDP, HTTP, FTP, and DNS, and understand their roles in data communication.
6. Explore OSI and TCP/IP Models and how data flows through different network layers.
7. Implement Routing and Switching Techniques including static and dynamic routing protocols

COURSE OUTCOME:

The students will be able to:

CO1:	Describe the general principles of data communication, the concept of the layered approach	K2
CO2:	Describe how computer networks are organized with the concept of layered approach	K2
CO3:	Design logical sub-address blocks with a given address block and network topology	K3
CO4:	Understanding of simple LAN with hubs, bridges, and switches	K2
CO5:	Describe how routing protocols work	K3
CO6:	Understand network security threats and basic security mechanisms to protect data and communication.	K2

COURSE CONTENT:

MODULE 1:		**10 Hours**
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.		

MODULE 2:		**12 Hours**
Data link layer: Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.		
MODULE 3:		**10 Hours**
Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.		
MODULE 4:		**8 Hours**
Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.		
MODULE 5:		**5 Hours**
Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	1	2	-	-
CO2	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO3	3	3	2	-	2	-	-	-	-	-	-	-	2	1	-
CO4	2	2	-	-	2	-	-	-	-	-	-	-	2	-	-
CO5	3	2	-	-	3	-	-	-	-	-	-	-	2	-	-
CO6	2	2	-	-	2	2	-	2	-	-	-	-	2	-	1
	2.66	2.166	2		2.25	2		2				1	2	1	1

Books:

1. A. S. Tanenbaum and D. J. Wetherall, “Computer Networks”, Pearson, 5th Edition, 2010, ISBN-10: 0132126958, ISBN-13: 978-0132126953.
2. B. A. Forouzan, “Data Communications and Networking”, McGraw-Hill Education, 5th Edition, 2012, ISBN-10: 0073376221, ISBN-13: 978-0073376226.
3. J. F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach”, Pearson, 8th Edition, 2021, ISBN-10: 0136681553, ISBN-13: 978-0136681557.
4. W. Stallings, “Data and Computer Communications”, Pearson, 10th Edition, 2013, ISBN-10: 0133506487, ISBN-13: 978-0133506488.

5. D. E. Comer, "Computer Networks and Internets", Pearson, 6th Edition, 2014, ISBN-10: 0133587932, ISBN-13: 978-0133587937.
6. M. A. Gallo and W. M. Hancock, "Computer Communications and Networking Technologies", Cengage Learning, 1st Edition, 2001, ISBN-10: 053437130X, ISBN-13: 978-0534371305.

Compiler Design (TIU-UCS-T320)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 6th
Course Title: Compiler Design	**Subject Code:** TIU-UCS-T320
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

1. To make the student aware about the basic concepts, i.e. different phases such as lexical analysis, syntax analysis, semantic analysis and code generation of compiler.
2. The students should know the different functionalities of compiler.
3. To make the students aware about the possible errors that can occur at different phases and how they can be addressed.
4. Make the students aware about the tools LEX and YACC.

COURSE OUTCOME:

The students will be able to:

CO1:	Understand fundamentals of language parser and identify the relationships among different phases of compiler	K2
CO2:	Illustrate the use of different types of parsers and their constructions, production rules and language semantics.	K3
CO3:	Inherited and synthesized attributes with their evaluations, run time storage allocation.	K3
CO4:	Describe techniques for intermediate code generation and code optimization.	K2
CO5:	Analyze error detection and recovery techniques in different compiler phases.	K3
CO6:	Implement and evaluate code generation techniques for efficiency.	K4

COURSE CONTENT:

MODULE 1:	**COMPILER STRUCTURE**	**3 Hours**
Analysis-synthesis model of compilation, various phases of a compiler, tool-based approach to compiler construction.		
MODULE 2:	**LEXICAL ANALYSIS**	**6 Hours**
Interface with input, parser and symbol table, token, lexeme and patterns, difficulties in lexical analysis, error reporting, and implementation. Regular definition, Transition diagrams, LEX		
MODULE 3:	**SYNTAX ANALYSIS**	**18 Hours**
Context free grammar, ambiguity, associativity, precedence, top-down parsing, recursive descent parsing, transformation on the grammars, predictive parsing, Bottom-up parsing, operator		

precedence grammars, LR parsers (SLR, LALR, LR), YACC.		
MODULE 4:	**SYNTAX DIRECTED DEFINITIONS**	**3 Hours**
Inherited and synthesized attributes, dependency graph, evaluation order, bottom up and top-down evaluation of attributes, L- and S-attributed definitions		
MODULE 5:	**TYPE CHECKING**	**2 Hours**
Type system, type expressions, structural and name equivalence of types, type conversion, overloaded functions and operators, polymorphic functions.		
MODULE 6:	**RUN TIME SYSTEM**	**2 Hours**
Storage organization, activation tree, activation record, parameter passing, Symbol table, dynamic storage allocation. Intermediate code generation: Intermediate representations, translation of declarations, assignments Intermediate Code generation for control flow, Boolean expressions and procedure calls, implementation issues.		
MODULE 7:	**CODE GENERATION AND INSTRUCTION SELECTION**	**6 Hours**
Issues, basic blocks and flow graphs, register allocation, code generation DAG representation of programs, code generation from DAGs, peep-hole optimization, code generator generators, specifications of machine		
MODULE 8:	**CODE OPTIMIZATION**	**5 Hours**
Source of optimizations, and optimization of basic blocks, loops, global dataflow analysis, and solution to iterative data flow equations. Code improving transformations, dealing with aliases, data flow analysis of structured flow graphs.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	-	-	-	-	-	-	-	-	-	1	2	-	-
CO2	3	3	-	-	1	-	-	-	-	-	-	-	2	-	-
CO3	3	2	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	3	2	-	-	1	-	-	-	-	-	-	-	2	-	-
CO5	2	3	-	-	-	-	-	-	-	-	-	-	1	-	-
CO6	3	3	-	-	1	-	-	-	-	-	-	-	2	-	-
	2.833	2.66			1							1	1.83		

Books:

1. Aho, Ullman,Sethi and Lam, Principles of Compiler Design, Pearson Education

2. Holub, Compiler Design in C, PHI
3. Andrew L. Appel, Modern Compiler Implementation in C, Foundation Books, Delhi
4. Dick Gruneet. Al., Modern Compiler Design, Wiley Dreamtech
5. S. Chattopadhyay, Compiler Design, PHIS. Pal: Systems Programming, Oxford University Press

Software Engineering (TIU-UCS-T314)

Program: B. Tech. in CSE	**Year, Semester**: 3rd, 6th
Course Title: Software Engineering	**Subject Code**: TIU-UCS-T314
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit**: 3

COURSE OBJECTIVE

1. To develop basic Knowledge in Software Engineering and its applications.
2. To understand software Engineering layered architecture and the process framework.
3. To analyze software process models such as the waterfall, spiral, evolutionary models and agile method for software development.
4. To design software requirements and specifications of documents.

COURSE OUTCOME

The students will be able to:

CO1:	Identify and describe fundamental software engineering concepts, principles and models.	K2
CO2:	Analyze and document software requirements using appropriate elicitation techniques and requirement engineering processes.	K4
CO3:	Design software solutions using modeling techniques such as UML, architectural styles, and design patterns.	K3
CO4:	Implement software applications by applying programming principles, coding standards, and development methodologies.	K3
CO5:	Evaluate software quality through testing strategies, verification, validation, and project management techniques.	K4
CO6:	Demonstrate teamwork, ethical considerations, and professional responsibility in software development projects.	K4

COURSE CONTENT

MODULE 1:	**FOUNDATIONS OF SOFTWARE ENGINEERING**	**12 Hours**
Introduction to software engineering: Software and software engineering, phases in software development, software development process models, role of management in software development, role of metrics and measurement.		
MODULE 2:	**SOFTWARE REQUIREMENTS AND PROJECT PLANNING**	**12 Hours**

Software requirement specifications: Role of SRS, problem analysis, requirement specification, validation, metrics, monitoring and control. Planning a software project: Cost estimation, project scheduling, staffing, personal planning, team structures, SCM, quality assurance plans, project-monitoring plans, risk management, Knowledge driven approach and development.		
MODULE 3:	**SYSTEM AND DETAILED DESIGN**	**7 Hours**
System design: Design objectives, design principles, module level concepts, design methodology, structured design, design specifications, verification metrics, monitoring and control. Detailed design: Module specification, detailed design and process design language, verification.		
MODULE 4:	**CODING AND TESTING**	**7 Hours**
Coding: Programming practice, verification, and metrics. Testing: Testing fundamentals, functional testing, structural testing, testing process, comparison of different V & V techniques.		
MODULE 5:	**SOFTWARE QUALITY AND RELIABILITY**	**7 Hours**
Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.		
	TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	1	-	1	3	-	-
CO2	2	3	-	1	-	-	-	-	-	1	-	-	2	-	-
CO3	2	2	3	-	1	-	-	-	-	1	-	-	2	-	1
CO4	2	2	3	-	1	-	-	-	-	-	-	-	2	-	2
CO5	2	2	1	3	1	-	-	-	-	-	1	-	2	-	1
CO6	-	-	-	-	-	1	-	1	3	2	2	-	1	-	-
	2.2	2.2	2.33	2	1	1		1	3	1.25	1.5	1	2		1.33

Books:

1. Roger S Pressman, Software Engineering-A Practitioners Approach, McGraw Hill Publications.
2. Pankaj Jalote, An Integrated Approach to Software Engineering, BPB Publications

3. Rajib Mall, Fundamentals of Software Engineering, PHI Learning Private Limited
4. Software Engineering, Ian Sommerville

Computer Networks Lab (TIU-UCS-L394)

Program: B.Tech. in CSE	**Year, Semester:** 3rd, 6th.
Course Title: Computer Networks Lab	**Subject Code:** TIU-UCS-L394
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Enumerate various network topologies and identify situations when different network topologies would be useful.
2. Explain and apply error control mechanisms to ensure reliable data transmission in computer networks.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Explain the error control mechanisms in computer network infrastructure.	K3
CO-2	Identify and describe the network layers, structure/format, and the role of each network layer.	K3
CO-3	Design and implement various network applications such as data transmission.	K4
CO-4	Illustrate the connectivity and data transmission between client and server in real-time multimedia transmission.	K3
CO-5	Distinguish and explain various routing protocols, algorithms, and internetworking mechanisms.	K4
CO-6	Evaluate and troubleshoot network performance, addressing issues related to bandwidth, latency, and network reliability in both local and wide-area networks.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO NETWORK**	**9 Hours**
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.		
MODULE 2:	**DATA LINK LAYER**	**10 Hours**
Data link layer: Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channels. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sublayer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link		

layer switching.		
MODULE 3:	**NETWORK LAYER**	**9 Hours**
Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.		
MODULE 4:	**TRANSPORT LAYER**	**9 Hours**
Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.		
MODULE 5:	**APPLICATION LAYER**	**8 Hours**
Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	-	1	2	-	-	-	-	1	-	1	2	-	-
CO2	2	2	-	1	2	-	-	-	-	-	-	1	2	-	-
CO3	2	2	3	2	3	-	-	-	-	1	-	1	3	-	-
CO4	2	2	3	2	3	-	-	-	-	1	-	1	3	-	-
CO5	2	3	2	2	2	-	-	-	-	-	-	1	3	-	-
CO6	2	2	2	3	3	-	-	-	-	-	1	2	3	-	1
	2	2.166	2.5	1.833	2.5					1	1	1.166	2.666		1

Books:

1. Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer Networks (5th ed.). Pearson.
2. Forouzan, B. A. (2017). Data Communications and Networking (5th ed.). McGraw-Hill Education.
3. Stallings, W. (2020). Data and Computer Communications (11th ed.). Pearson.
4. Comer, D. E. (2018). Computer Networks and Internets (6th ed.). Pearson.
5. Kurose, J. F., & Ross, K. W. (2021). Computer Networking: A Top-Down Approach (8th ed.). Pearson.
6. Peterson, L. L., & Davie, B. S. (2021). Computer Networks: A Systems Approach (6th ed.). Morgan Kaufmann.

Software Engineering Lab (TIU-UCS-L352)

Program: B. Tech. in CSE	**Year, Semester:** 3rd, 6th
Course Title: Software Engineering Lab	**Subject Code:** TIU-UCS-L352
Contact Hours/Week: 0–0–3 (L–T–P)	**Credit:** 1.5

COURSE OBJECTIVE

Enable the student to:

1. Analyze software process models such as the waterfall, spiral, evolutionary models and agile method for software development.
2. Design software requirements and specifications of documents, project planning, scheduling, cost estimation, risk management.
3. Describe data models, object models, context models, behavioral models, coding style and testing issues. Also to know about the quality checking mechanism for software processes and products.

COURSE OUTCOME

On completion of the course, the student will be able:

CO1	Identify and examine requirements from problem statements to construct software solutions.	K3
CO2	Develop and design software solutions using UML modeling techniques.	K4
CO3	Illustrate and apply software engineering principles to organize and manage software projects effectively.	K4
CO4	Test and validate software systems using appropriate testing strategies and tactics.	K3
CO5	Implement and assess software metrics to improve the quality and maintainability of software products.	K4
CO6	Demonstrate ethical, social, and legal responsibilities in software development and ensure compliance with industry standards.	K3

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO SOFTWARE ENGINEERING**	**9 Hours**
Introduction to software engineering: Software and software engineering, phases in software development, software development process models, role of management in software development, role of metrics and measurement.		
MODULE 2:	**REQUIREMENT ANALYSIS AND SPECIFICATION, PROJECT MANAGEMENT**	**12 Hours**
Software requirement specifications: Role of SRS, problem analysis, requirement specification, validation, metrics, monitoring and control. Planning a software project: Cost estimation, project scheduling, staffing, personal planning, team structures, SCM, quality assurance plans, project-monitoring plans, risk management, Knowledge driven approach and development.		
MODULE 3:	**SOFTWARE DESIGN**	**9 Hours**
System design: Design objectives, design principles, module level concepts, design methodology, structured design, design specifications, verification metrics, monitoring and control. Detailed design: Module specification, detailed design and process design language, verification.		
MODULE 4:	**CODING AND TESTING**	**9 Hours**

Coding: Programming practice, verification, and metrics. Testing: Testing fundamentals, functional testing, structural testing, testing process, comparison of different V & V techniques.		
MODULE 5:	**SOFTWARE QUALITY**	**6 Hours**
Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	-	1	2	-	-	-	-	1	-	1	2	-	-
CO2	2	2	3	-	3	-	-	-	-	-	-	1	3	-	1
CO3	2	2	3	1	2	-	-	2	1	1	1	2	3	-	-
CO4	2	2	2	2	3	-	-	-	-	1	-	1	3	-	1
CO5	2	2	2	2	3	-	-	-	-	-	-	1	3	-	1
CO6	-	-	-	-	-	2	1	3	2	2	2	1	2	-	-
	2	2	2.5	1.5	2.6	2	1	2.5	1.5	1.25	1.5	1.166	2.66		1

Books:

1. Software Engineering, Ian Sommerville
2. R. Mall, "Fundamentals of Software Engineering", Prentice Hall of India
3. R. S. Pressman, "Software Engineering: A Practitioner's Approach", Tata McGraw Hill
4. D. Bell, "Software Engineering for Students", Pearson

Compiler Design Lab (TIU-UCS-L396)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Compiler Design Lab	**Subject Code:** TIU-UCS-L396
Contact Hours/Week: 0–0–3 (L–T–P)	**Credit:** 1.5

COURSE OBJECTIVE:

1. Make a student familiar with tools related to lexical and syntax analysis.
2. A student can generate lexical analyzers and parsers.
3. Make a student enable to create compiler for new languages.

COURSE OUTCOME:

After completion of the lab course, A student should be able to,

CO1	Familiar with the tools LEX and YACC.	K2
CO2	Can implement programs using **LEX**.	K3
CO3	Can implement programs using **YACC**.	K3
CO4	Generate lexical analyzer using **LEX**.	K3
CO5	Generate parser using **YACC**.	K3
CO6	Able to create compilers for a new language.	K6

COURSE CONTENT:

MODULE 1:	**Introduction to LEX and YACC**	**6 Hours**
- Install and configure LEX and YACC - Overview of LEX and YACC - Introduction to lexical analysis and syntax parsing		
MODULE 2:	**Basic LEX Programs**	**6 Hours**
- LEX Program to count the number of words, spaces, and lines - LEX Program to calculate factorial of a number - LEX Program to print the table of a number		
MODULE 3:	**Advanced LEX Programs**	**9 Hours**
- LEX Program to identify and count positive and negative numbers - LEX Program to convert lowercase to uppercase and reverse - LEX Program to accept a string starting with a vowel - LEX Program to find if a character apart from alphabets occurs in a string		
MODULE 4:	**LEX Program for String and Number Manipulations**	**9 Hours**
- LEX Program to check if a string is palindrome - LEX Program to calculate sum of the digits of a number - LEX Program to check if a number is palindrome		
MODULE 5:	**Advanced YACC Programs and User-defined Language**	**15 Hours**

- YACC Program to implement a calculator - YACC Program to evaluate arithmetic expressions - YACC Program to implement LL (1) parser - YACC Program to implement SLR parser - Designing a user-defined language and generating corresponding lexical and syntactic rules for the lexical analyzer and parser

TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	2	2	1	3	2	3	2	3	3	2
CO2	3	3	3	2	3	3	2	1	2	3	2	3	3	2	3
CO3	3	3	3	2	3	3	2	1	2	3	2	3	3	2	3
CO4	3	3	3	3	3	3	2	1	3	3	3	3	3	3	2
CO5	3	3	3	3	3	3	3	1	3	3	3	3	3	3	3
CO6	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3
	3	2.83	2.5	3	2.83	2.33	1.17	2.67	2.83	2.83	2.67	2.83	3	2.83	2.5

Books:

1. Tom Niemann. LEX & YACC TUTORIAL

Web Technology (TIU-UCS-E322)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Web Technology	**Subject Code:** TIU-UCS-E322
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

1. Understand Windows Concepts and Programming.
2. Master Dynamic and Active Web Pages.
3. Learn Java Web Development Technologies.
4. Understand the Basics of Web Services and Enterprise JavaBeans (EJB), Java Server Pages.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO1	Understand the concepts of Internet Infrastructure.	K2
CO2	To introduce the fundamentals of Internet, and the principles of web design.	K2
CO3	Implement the concepts of client side scripting language.	K3
CO4	To construct basic websites using HTML and Cascading Style Sheets	K3
CO5	To build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms, come security aspects of communication.	K4
CO6	Develop solution to complex problems using appropriate method and use of JSP in web technology related activities.	K4

COURSE CONTENT:

MODULE 1:	**WINDOWS CONCEPTS AND TERMINOLOGY**	**8 Hours**
Key elements, Creating the look, communication via messages, windows resources and functions, adding multimedia and sound Resources Writing windows applications, taking control of windows, adding menus, dialog boxes, Special controls. Concepts of X-Windows System & programming.		
MODULE 2:	**DYNAMIC WEB PAGES**	**12 Hours**
The need of dynamic web pages; an overview of DHTML, cascading style sheet (css), comparative studies of different technologies of dynamic page creation.		
MODULE 3:	**ACTIVE WEB PAGES**	**7 Hours**
Need of active web pages; java applet life cycle. Java Script, Data types, variables, operators, conditional statements, array object, date object, string object. Java Servlet, Servlet environment and role, HTML support, Servlet API, The servlet life cycle, Cookies and Sessions.		
MODULE 4:	**JSP**	**10 Hours**
JSP architecture, JSP servers, JSP tags, understanding the layout in JSP, Declaring variables, methods in JSP, inserting, java expression in JSP, processing request from user and generating dynamic response for the user, inserting applets and java beans into JSP, using include and forward action, comparing JSP and CGI program, comparing JSP and ASP program; Creating ODBC data source name, introduction to JDBC, prepared statement and callable statement.		
MODULE 5:	**J2EE and COMMUNICATION ASPECTS**	**8 Hours**
An overview of J2EE web services, basics of Enterprise Java Beans, EJB vs. Java Beans, basics of RMI, JNI		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	1	1	2	3	1	2	1	2	3	2	2
CO2	3	3	3	2	2	1	2	2	1	2	1	2	3	2	3
CO3	2	3	3	3	3	2	3	2	1	3	2	3	2	3	3
CO4	3	2	3	3	1	1	2	2	1	2	1	2	3	2	2
CO5	3	3	3	3	3	2	3	3	1	3	2	3	3	3	3
CO6	3	2	3	3	2	2	3	3	1	3	2	3	3	2	2
	2.83	2.67	2.67	2.5	2.33	1.83	2.33	2.17	1.17	2.5	1.83	2.5	2.83	2.5	2.5

Recommended Books:

Main Reading

1. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, ASP.NET, XML and Ajax, Dreamtech Press; first edition.
2. Web Technologies, Godbole and Kahate, Tata McGraw-Hill Education.
3. Web Technologies: A Computer Science Perspective, Jeffrey C. Jackson, Pearson,2011

Supplementary Reading

1. Web Technology: A Developer's Perspective, N.P.Gopalan and J. Akilandeswari, PHI Learning, Delhi, 2013.
2. Internetworking Technologies, An Engineering Perspective, Rahul Banerjee, PHI Learning, Delhi, 2011.
3. Java Servlets and JSP, Murach's.
4. Java for the Web with Servlets, JSP, EJB, Budi. Kurniawan.
5. Cryptography and Network security, William Stallings.

Computer Graphics (TIU-UCS-E328)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Computer Graphics & Multimedia Systems	**Subject Code:** TIU-UCS-E328
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3
Prerequisite Course: Data Structures and Algorithms (TIU-UCS-T201); Design and Analysis of Algorithms (TIU-UCS-T321); Discrete Mathematics (TIU-UMA-T215)	

COURSE OBJECTIVE:

1. To provide students with a strong foundation in computer graphics concepts, including coordinate systems, transformations, and rendering techniques.

2. To enable students to implement fundamental graphics algorithms such as line drawing, polygon filling, clipping, and 3D transformations using programming languages like C or C++.
3. To equip students with the skills to develop computer graphics applications in fields such as gaming, scientific visualization, and virtual reality.
4. To introduce students to advanced topics like shading models, texture mapping, and animation while familiarizing them with industry-standard graphics tools and libraries.

COURSE OUTCOME:

CO No.	Course Outcome (CO)	Bloom's Taxonomy Level
CO1	Explain the basic concepts of computer graphics, including coordinate representation, graphics output devices, and input devices.	K2
CO2	Demonstrate an understanding of 2D and 3D transformations, viewing, and projections in computer graphics.	K3
CO3	Implement fundamental graphics algorithms such as line drawing, polygon filling, and clipping using C or C++.	K3
CO4	Apply rendering techniques like scan-line, illumination models, and shading for realistic image generation.	K4
CO5	Develop computer graphics solutions for applications such as gaming, scientific visualization, and virtual reality using curves, surfaces, and color models.	K6
CO6	Analyze and apply computer graphics techniques like animation, modeling, and texture mapping for real-world problem-solving.	K5

COURSE CONTENT:

MODULE 1:	Introductory Concepts	10 Hours
Introduction to Computer Graphics and Applications, Coordinate Representation and Pixel Graphics, Graphics Output Devices: CRT, Raster Scan, Random Scan, Color CRT Monitors, DVST, Flat Panel Displays, Video Controller and Raster Scan Display Processor, Graphics Input Devices (Mouse, Keyboard, Joystick, etc.), Graphics Software and Frameworks, Summary and Discussion on Introductory Concepts		

MODULE 2:	Graphics Output Primitives	10 Hours
Point and Line Representation, DDA Line Drawing Algorithm, Bresenham's Line Drawing Algorithm, Circle and Ellipse Drawing Algorithms, Polygon Drawing and Representation, Scan Conversion and Real-time Rendering, Run-length Encoding and Character Generation, Anti-aliasing Techniques, Review and Practice Problems on Output Primitives		
MODULE 3:	**2D Viewing**	**7 Hours**
Viewing Pipeline and Window-to-Viewport Transformation, 2D Clipping: Cohen-Sutherland Algorithm, Midpoint Subdivision Algorithm, Liang-Barsky Line Clipping, Cyrus-Beck Line Clipping, Polygon Clipping: Sutherland-Hodgeman Algorithm, Weiler-Atherton Polygon Clipping, Character Clipping and Summary		
MODULE 4:	**2D and 3D Transformations**	**8 Hours**
Basic 2D Transformations: Scaling, Rotation, Translation, Shearing, Reflection and Composite Transformations, Homogeneous Coordinates and Affine Transformations, 3D Transformations: Translation, Scaling, Rotation, Solid Body Transformations and Projections, Perspective and Orthographic Projections, Axonometric and Oblique Projections, Review and Problem Solving on Transformations		
MODULE 5:	**Advanced Topics**	**10 Hours**
Curves and Surfaces: Spline Representations, Bezier Curves and Surfaces, B-Spline Curves and Surfaces, Visible Surface Detection: Back-face Detection, Depth-Buffer, A-Buffer, and Z-Buffer Techniques, Illumination Models and Surface Rendering, Color Models and Half-toning Techniques, Applications in Game Development and VR, Summary and Future Trends in Computer Graphics		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	1	1	2	3	1	2	1	2	3	2	2
CO2	3	3	3	2	2	1	2	2	1	2	1	2	3	2	3
CO3	2	3	3	3	3	2	3	2	1	3	2	3	2	3	3

CO4	3	2	3	3	1	1	2	2	1	2	1	2	3	2	2
CO5	3	3	3	3	3	2	3	3	1	3	2	3	3	3	3
CO6	3	2	3	3	2	2	3	3	1	3	2	3	3	2	2
	2.83	2.67	2.67	2.5	2.33	1.83	2.33	2.17	1.17	2.5	1.83	2.5	2.83	2.5	2.5

Recommended Books:

Main Reading

1. D. Hearn and P M Baker, Computer Graphics – C Version, Prentice Hall of India, 4th Edition, 2014.
2. Z. Xiang, Roy Plastock, Computer Graphics, Schaum's Outline series, McGraw-Hill, 2nd Edition, 2000.
3. A. Mukhopadhyay, A Chattopadhyay, Introduction to Computer Graphics and Multimedia, Vikas Publishing House, 2nd Edition, 2013.
4. D. F. Rogers, J A Adams, Mathematical Elements for Computer Graphics, Tata McGrow-Hill, 2nd Edition.

Supplementary Reading

1. J F Hughes, A Van Dam, M Mcguire, D F Sklar, J D Foley, S K Feiner, K Akeley, Computer Graphics – Principles and Practice, 3rd Edition, Pearson, 2013
2. Edward Angel, Dave Shreiner, Interactive Computer Graphics – A top-down Approach with Shader-based OpenGL, 6th Edition, Pearson, 2012.
3. Frank Klawonn, Introduction to Computer Graphics – Using Java 2D and 3D, Springer, 2008.

Data Analytics (TIU-UCS-E326)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Data Analytics	**Subject Code:** TIU-UCS-E326
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

1. Make a student familiar with the basics of inferential statistics.
2. Make a student familiar with different python tools.
3. Make a student familiar with different machine learning methods for data analysis.
4. A student should be able to apply different machine learning methods for data analysis.
5. A student should be able to analyze unstructured data with text mining.

COURSE OUTCOME:

After the completion of the course, a student should be able to:

CO1	Understand and apply inferential statistics	K2, k3
CO2	Understand and apply different python libraries used to analysis data	K2, k3
CO3	Understand and apply different machine learning algorithms for given data	K2, k3

CO4	Evaluate the performance of different machine learning methods for given data	K4
CO5	Understand and apply ensemble models	K2, k3
CO6	Understand and apply methods over unstructured text	K2, k3

COURSE CONTENT:

MODULE 1:	**Inferential Statistics**	**11 Hours**
Various forms of distribution: Normal distribution, Poisson distribution, Bernoulli distribution; Basic Analysis Techniques: Statistical hypothesis generation and testing; Chi-Square test; chi-square test of independence; z-score; T-test; Z-test vs T-test; Analysis of variance (ANOVA); Correlation analysis; Maximum likelihood test.		
MODULE 2:	**Work with data in Python**	**8 Hours**
The world of arrays with NumPy Creating an array, Mathematical operations, Array subtraction, Squaring an array, A trigonometric function performed on the array, Conditional operations, Matrix multiplication, Indexing and slicing, Shape manipulation Empowering data analysis with pandas The data structure of pandas: Series, Data Frame, Panel, Inserting and exporting data.		
MODULE 3:	**Data Analysis by Machine Learning**	**11 Hours**
Different types of machine learning: Supervised learning, Unsupervised learning, Reinforcement learning; Data analysis techniques: Introduction to Regression, Classification and Clustering; Linear regression; Logistic regression; The naive Bayes classifier; Decision trees; The K-nearest neighbour; The k-means clustering.		
MODULE 4:	**Pushing Boundaries with Ensemble Models**	**7 Hours**
Basics of Ensemble Methods, Types of Ensemble Methods: Voting and Averaging Based Ensemble Methods, Stacking, Bagging, Boosting, Random forests.		
MODULE 5:	**Analyzing Unstructured Data with Text Mining**	**8 Hours**
Preprocessing data; Word and sentence tokenization; Parts of speech tagging; Stemming and lemmatization; The Stanford Named Entity Recognizer; Performing sentiment analysis.		
TOTAL LECTURES		**45 Hours**

.

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	1	2	3	2	2	1	2	3	2	2	3	2	3
CO2	2	3	2	3	3	3	2	2	3	2	3	2	2	3	2
CO3	3	3	3	3	2	3	2	2	3	3	2	2	3	3	3

CO4	2	2	3	3	3	3	3	2	3	2	3	3	3	3	2
CO5	3	3	3	2	3	3	2	2	3	3	3	2	3	3	3
CO6	2	3	2	3	2	3	3	3	2	3	2	2	3	2	3
	2.5	2.5	2.5	2.5	2.7	2.5	2.3	2.2	2.5	2.5	2.5	2.5	2.6	2.6	2.6

Recommended Books:

Main Reading

1. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data,O'Reilly
2. Samir Madhavan, Mastering Python for Data Science
3. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye, Prentice Hall Inc.

Supplementary Reading

1. Alberto Boschetti, Luca Massaron, Python Data Science Essentials

Image Processing (TIU-UCS-E330A)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Image Processing	**Subject Code:** TIU-UCS-E330A
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

1. To introduce the fundamentals of digital image processing including image representation, sampling, quantization, and the basic building blocks of an image processing system.
2. To equip students with knowledge of image enhancement and transformation techniques in both spatial and frequency domains, enabling them to improve image quality and extract relevant features.
3. To develop the ability to analyze and implement various image restoration, segmentation, compression, and morphological operations, facilitating better understanding of image manipulation and interpretation.
4. To foster the application of advanced image processing techniques, such as wavelet transforms and object recognition, for real-world applications in computer vision, medical imaging, and machine learning.

COURSE OUTCOME:

The students will be able to:

CO1	Understand the fundamental concepts of digital image processing, including sampling and quantization, image transforms, and image enhancement.	K2
CO2	Apply spatial and frequency domain methods to enhance images.	K3
CO3	Segment images using edge detection, thresholding, and region-based methods.	K3
CO4	Represent and describe images using different schemes.	K2
CO5	Understand the fundamental problems in pattern recognition, including classification, clustering, and feature selection.	K2
CO6	Implement and evaluate image processing and pattern recognition techniques in real-world applications.	K4

COURSE CONTENT:

Module No.	Module Topic	Total Hours
Module 1	Introduction and Digital Image Fundamentals	6 hours
The origins of Digital Image Processing, Application domain of Image Processing, Fundamental Steps in Image Processing, Elements of Digital Image Processing Systems, Image Sampling and Quantization, Pixel connectivity, Distance measures between pixels, Translation, Scaling, Rotation and Perspective Projection of image, Linear and Non-Linear Operations, Image types, Image representation in memory.		
Module 2	Intensity Transformations and Spatial Filtering	5 hours
Basic gray-level transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Combining Spatial Enhancement Methods, Spatial Filters, Smoothening and Sharpening Spatial Filters.		
Module 3	Image Processing in Frequency and Wavelet Domain	6 hours
Introduction to Fourier Transform and the frequency Domain, Computing and Visualizing the 2D DFT, Smoothing using Frequency Domain Filters, Sharpening in Frequency Domain, Homomorphic Filtering, Concept of multi-resolution image processing, 2D wavelet transformation.		
Module 4	Image Restoration	5 hours
A model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering Processing Application, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Geometric Mean Filter, Geometric Transformations.		
Module 5	Image Segmentation	4 hours
Point-line-edge detection, Detection of Discontinuities, Edge linking and boundary detection, Thresholding, Region-based segmentation.		
Module 6	Image Compression and Color Image Processing	4 hours

Coding, Inter-pixel and Psycho-visual Redundancy, Image Compression models, Compression standards – JPEG compression, Color image representation, Conversion to various color spaces.		
Module 7	Morphological Image Processing	5 hours
Basic concepts of set theory, Basic morphological operations, Image filtering using morphological operations.		
Module 8	Object Recognition	10 hours
Patterns and Pattern Classes, Decision-Theoretic Methods, Structural Methods.		
Total Hours:		45 hours

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	2	3	2	2	1	2	3	2	2	3	2	3
CO2	2	3	2	3	3	3	2	2	3	2	3	2	2	3	2
CO3	3	3	3	3	2	3	2	2	3	3	2	2	3	3	3
CO4	2	2	3	3	3	3	3	2	3	2	3	3	3	3	2
CO5	3	3	3	2	3	3	2	2	3	3	3	2	3	3	3
CO6	2	3	2	3	2	3	3	3	2	3	2	2	3	2	3
	2.5	2.5	2.5	2.5	2.7	2.5	2.3	2.2	2.5	2.5	2.5	2.5	2.6	2.6	2.6

Recommended Books:

Main Reading

1. R. C. Gonzalez, W. E. Woods, “Digital Image Processing”, Pearson-Prentice Hall, Eds. – 3, 2008

Supplementary Reading

1. A. K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, Addison-Wesley, 1989
2. W. K. Pratt, “Digital Image Processing”, PIKS inside, Wiley, Newyork, 3rd Eds, 2001
3. M. Petrou, P. Bosdogianni, “Image Processing- The Fundamentals”, Wiley, 1999
4. Al Bovik, “Handbook of Image and Video Processing”, Academic Press, 2000
5. J. C. Russ, “The Image Processing Handbook”, CRC, Boca Raton, FL, 4th Edition, 2002

Image Processing Lab (Elective - II Lab) (TIU-UCS-L330A)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Image Processing Lab	**Subject Code:** TIU-UCS-L330A
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

1. To enable students to understand the fundamental concepts of digital image processing and apply techniques such as image enhancement, restoration, and segmentation in practical scenarios.
2. To provide hands-on experience in implementing various image processing operations using tools such as MATLAB or Python (OpenCV), fostering the ability to design, analyze, and test image-based algorithms.
3. To cultivate the ability to relate theoretical principles of image processing to real-world applications in areas like medical imaging, computer vision, surveillance, and pattern recognition.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO No.	Course Outcome Statement	Bloom's Level
CO1	Recall and explain the basic concepts of digital image representation, sampling, and quantization.	K1
CO2	Interpret image histograms and apply basic image enhancement and filtering techniques.	K2
CO3	Implement spatial and frequency domain processing techniques using suitable software tools.	K3
CO4	Analyze the effects of different restoration, segmentation, and morphological operations.	K4
CO5	Evaluate performance of image compression and object recognition methods on test datasets.	K5
CO6	Design and create end-to-end image processing applications for real-world problems.	K6

COURSE CONTENT:

MODULE 1:	**Introduction and Digital Image Fundamentals**	**6 Hours**

Image loading and saving, Image display, Color to grayscale conversion, Image sampling, Image quantization, Pixel connectivity, Distance measures (Euclidean, City-block, Chessboard), Image translation, Image scaling, Image rotation, Perspective projection, Linear operations, Non-linear operations, Image types, Image memory representation		
MODULE 2:	**Intensity Transformations and Spatial Filtering**	**7 Hours**
Gray-level transformations (negative, log, gamma), Histogram computation, Histogram equalization, Image enhancement using arithmetic operations, Image enhancement using logic operations, Combining spatial enhancement methods, Smoothing filters (mean, Gaussian, median), Sharpening filters (Laplacian, Sobel)		
MODULE 3:	**Image Processing in Frequency and Wavelet Domain**	**6 Hours**
2D Fourier Transform computation, Fourier spectrum visualization, Frequency domain low-pass filtering, Frequency domain high-pass filtering, Homomorphic filtering, Concept of multiresolution analysis, 2D Wavelet Transform, Wavelet decomposition and reconstruction		
MODULE 4:	**Image Restoration**	**7 Hours**
Image degradation model, Noise models (Gaussian, Salt & Pepper), Noise addition, Spatial restoration using mean filter, Spatial restoration using median filter, Periodic noise removal in frequency domain, Inverse filtering, Wiener filtering, Estimation of degradation function, Geometric mean filtering, Geometric transformations		
MODULE 5:	**Image Segmentation**	**5 Hours**
Point detection, Line detection, Edge detection (Sobel, Prewitt, Canny), Detection of discontinuities, Edge linking, Boundary detection, Global thresholding, Adaptive thresholding, Otsu's method, Region growing, Region splitting, Region merging		
MODULE 6:	**Image Compression and Color Image Processing**	**4 Hours**
Image compression basics, Inter-pixel redundancy, Psycho-visual redundancy, Run-length encoding, JPEG compression overview, Color image representation, RGB to HSV conversion, RGB to YCbCr conversion, Grayscale conversion, Color channel separation and visualization		
MODULE 7:	**Morphological Image Processing**	**4 Hours**
Set theory basics, Binary image representation, Erosion, Dilation, Opening, Closing, Morphological gradient, Noise removal using morphology, Boundary extraction using morphological operations		
MODULE 8:	**Object Recognition**	**6 Hours**
Pattern classes, Feature extraction (area, perimeter, centroid), Shape descriptors, Decision-theoretic classification, k-NN classification, Structural representation, Template matching, Graph-based object representation		
	TOTAL LAB HOURS	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	-	1	3	2	1
CO2	3	2	-	-	3	-	-	-	-	-	-	-	3	2	1
CO3	3	3	2	2	-	-	-	-	1	-	-	-	3	3	2
CO4	3	2	-	2	3	-	-	-	1	-	-	1	3	3	2
CO5	2	1	2	-	3	-	-	-	-	-	-	1	2	2	1
CO6	3	2	-	2	3	-	-	-	1	-	-	2	3	3	2
	2.833	2	2	2	2.8				1			1.25	2.83	2.5	1.5

Books:

Main Reading

1. R. C. Gonzalez, W. E. Woods, “Digital Image Processing”, Pearson-Prentice Hall, Eds. – 3, 2008

Supplementary Reading

1. A. K. Jain, “Fundamentals of Digital Image Processing”, Prentice Hall, Addison-Wesley, 1989
2. W. K. Pratt, “Digital Image Processing”, PIKS inside, Wiley, Newyork, 3rd Eds, 2001
3. M. Petrou, P. Bosdogianni, “Image Processing- The Fundamentals”, Wiley, 1999
4. Al Bovik, “Handbook of Image and Video Processing”, Academic Press, 2000
5. J. C. Russ, “The Image Processing Handbook”, CRC, Boca Raton, FL, 4th Edition, 2002

Web Technology Lab (Elective - II Lab) (TIU-UCS-L330B)

Program: B. Tech in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Web Technology	**Subject Code:** TIU-UCS-L330B
Contact Hours/Week: 0-0-3 (L-T-P)	**Credit:** 1.5

COURSE OBJECTIVE:

1. Understand and apply concepts of Windows and web programming.
2. Gain hands-on experience in creating dynamic, interactive websites.
3. Learn Java-based server-side technologies like JSP and Servlets.
4. Explore advanced web technologies like J2EE and web services.

COURSE OUTCOME:

Upon completion of the course, the student will be:

CO-1	Design and implement Windows applications using key elements of Windows concepts, including multimedia integration, and user interface controls.	K1,K 2
CO-2	Develop dynamic web pages using DHTML and CSS to create interactive websites that adapt to user inputs and improve user experience.	K3
CO-3	Build active web pages with JavaScript and Java applets to enhance user interaction, managing data types, operators, arrays, and conditional logic.	K3,K 4
CO-4	Write and deploy Java Servlets to handle HTTP requests, manage sessions and cookies, and understand the Servlet lifecycle to develop robust server-side applications.	K5
CO-5	Create dynamic web content using JSP, including data handling, integrating Java Beans, applets, and working with database connectivity through JDBC.	K1, K5
CO-6	Implement J2EE technologies such as EJB and RMI to develop scalable, distributed, and secure enterprise applications, gaining proficiency in web services and remote communications.	K4, K6

COURSE CONTENT:

MODULE 1:	**Basics of Web Technology**	**10 Hours**
Introduction to web technologies and commonly used protocols: HTTP, HTTPS, TELNET, SMTP, POP3 Evolution and characteristics of WWW Understanding types of web pages: static, dynamic, and active web pages HTML: Elements, Attributes, Tables, Lists, Forms, Layouts CSS: Styling, Formatting, Layout using Inline, Internal, and External stylesheets Basics of IFrames, Colors, and Image Maps		
MODULE 2:	**Client-Side Scripting and Web Page Interaction**	**10 Hours**

JavaScript Basics: Data types, variables, operators, conditional statements JavaScript Objects: Arrays, Date, String JavaScript for form validation and DOM manipulation Introduction to XML: Syntax, Elements, Attributes, DTD, Schema validation Introduction to XHTML and its difference from HTML		
MODULE 3:	**Server-Side Programming using Java**	**10 Hours**
Java Servlet Basics: Lifecycle, request/response handling, HTML support Sessions and Cookies in Servlets JSP Fundamentals: architecture, directives, scriptlets, expressions, and declarations Processing user requests and generating dynamic responses using JSP Use of include and forward actions in JSP		
MODULE 4:	**Database Connectivity and Enterprise Technologies**	**8 Hours**
JDBC: Connecting to databases, Prepared and Callable statements Creating ODBC Data Source Names (DSN) Integration of JSP with JavaBeans and JDBC Overview of J2EE technologies: EJB, RMI Web services basics and remote communication		
MODULE 5:	**Security in Web Applications**	**7 Hours**
Network Security Threats: Malware (viruses, worms, trojans), Active and Passive attacks Network Security Techniques: Authentication, Password protection, VPN, IP Security Securing web applications: SSL, electronic transaction security Firewall types and working: Packet filtering, Stateful, Proxy		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	2	3	2	2	3	2	3	2	3	3	3	2
CO2	3	3	2	2	3	3	2	2	3	2	3	3	3	2	3
CO3	2	3	3	3	3	3	2	2	3	3	2	3	3	3	3
CO4	3	3	3	3	3	3	3	2	3	3	3	3	3	2	3
CO5	3	3	3	3	3	3	3	3	2	2	3	3	3	3	2

CO6	3	3	3	3	3	3	3	2	3	2	3	3	3	3	3
	2.83	2.83	2.83	2.66	3	2.83	2.5	2.33	2.66	2.5	2.66	3	3	2.66	2.66

Books:

Textbooks:

1. Clint Eccher, "Professional Web Design: Techniques and Templates (CSS & XHTML)",
2. Uttam K. Roy, "WEB TECHNOLOGIES".

Reference Books:

1. Jennifer Kyrnin Laura Lemay, Rafe Colburn, "Mastering HTML, CSS & JavaScript Web Publishing – 2023".
2. Kevin Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012, ISBN-10: 0262018020, ISBN-13: 978-0262018029
3. Godbole, "Web Technologies".

Computer Graphics Lab (Elective - II Lab) (TIU-UCS-L330C)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Computer Graphics Lab	**Subject Code:** TIU-UCS-L330C
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

1. **To provide a fundamental understanding of computer graphics concepts**, including image representation, visualization techniques, and the working principles of various display and input/output devices.
2. **To equip students with practical skills in implementing core graphics algorithms**, such as scan conversion, 2D and 3D transformations, clipping, and curve modeling using programming languages like C/C++ or Python.
3. **To develop the ability to design and render realistic graphical scenes**, by applying concepts of hidden surface removal, shading models, ray tracing, and texture mapping in both 2D and 3D environments.

COURSE OUTCOME:

CO-1	Describe the basic components of a computer graphics system and their applications in visualization and image processing.	K1

CO-2	Explain and analyze the working of scan conversion algorithms for drawing geometric primitives such as lines, circles, and ellipses.	K2, K4
CO-3	Apply 2D and 3D transformation techniques using matrix operations to manipulate geometric objects in space.	K3
CO-4	Demonstrate clipping algorithms and coordinate transformations for mapping objects from world coordinates to device coordinates.	K3
CO-5	Construct Bezier and B-spline curves for modeling smooth shapes and evaluating their behavior under different control points.	K6, K5
CO-6	Implement hidden surface removal, shading, and ray tracing algorithms to create realistic 3D graphics scenes.	K3, K6

COURSE CONTENT:

MODULE 1:	**Introduction to Computer Graphics**	**4 Hours**
Lab 1.1: Introduction to RGB Color Model Create a color mixer program (input RGB values and display the resulting color). **Lab 1.2: Visualization using Lookup Tables** Implement grayscale image enhancement using a lookup table. **Lab 1.3: Exploring Display Technologies** Prepare a presentation comparing raster scan and storage tube displays. **Lab 1.4: Active vs Passive Devices** Document use cases of input/output devices like mouse, keyboard, joystick, etc.		
MODULE 2:	**Scan Conversion**	**6 Hours**
Lab 2.1: DDA Line Drawing Algorithm Implement DDA and draw various lines. **Lab 2.2: Bresenham's Line Algorithm** Compare it with DDA on performance and accuracy. **Lab 2.3: Circle and Ellipse Generation** Implement midpoint circle and ellipse drawing. **Lab 2.4: Scan-line Polygon Fill Algorithm** Fill a convex polygon using scan-line method. **Lab 2.5: Flood Fill and Boundary Fill** Use recursion and stack-based techniques.		
MODULE 3:	**2D Transformation & Viewing**	**15 Hours**

Lab 3.1: 2D Transformation using Matrix Multiplication Create a program to input polygon vertices and apply transformations. **Lab 3.2: Composite Transformation** Combine transformations (e.g., rotate-then-translate). **Lab 3.3: Reflection and Shear** Implement these using homogeneous coordinates. **Lab 3.4: Cohen-Sutherland Line Clipping** Clip lines within a rectangular window. **Lab 3.5: Window to Viewport Mapping** Implement scaling and mapping logic.		
MODULE 4:	**3D Transformation & Viewing**	**6 Hours**
Lab 4.1: 3D Object Representation Model a cube or pyramid using wireframes. **Lab 4.2: 3D Transformations** Implement 3D translation, rotation (about axis), and scaling. **Lab 4.3: Arbitrary Axis Rotation** Rotate object around arbitrary axis using matrix multiplication. **Lab 4.4: 3D Clipping** Simulate clipping using bounding volumes. **Lab 4.5: Parallel Projection** Convert a 3D object to a 2D parallel projection.		
MODULE 5:	**Curves**	**4 Hours**
Lab 5.1: Bezier Curve Drawing Implement De Casteljau's algorithm. **Lab 5.2: B-Spline Curves** Draw B-spline with 4 control points. **Lab 5.3: Rational B-spline** Add weights and observe changes. **Lab 5.4: Curve Continuity** Experiment with end conditions for periodic B-splines.		
MODULE 6:	**Hidden Surface Removal**	**4 Hours**
Lab 6.1: Z-buffer Algorithm Create a simple depth buffer for overlapping triangles. **Lab 6.2: Back-face Detection** Cull polygons based on normal vector. **Lab 6.3: BSP Tree Visualization** Represent a basic binary space partitioning tree. **Lab 6.4: Painter's Algorithm** Implement basic back-to-front sorting and rendering.		
MODULE 7:	**Color & Shading Model**	**6 Hours**

Lab 7.1: Phong and Gouraud Shading Implement basic lighting models. **Lab 7.2: Light & Color Model Simulation** Simulate diffuse, ambient, and specular lighting. **Lab 7.3: Ray Tracing Basics** Trace rays to render spheres or planes with reflection. **Lab 7.4: Texture Mapping** Apply 2D image as a texture over a 3D model.
TOTAL LAB HOURS — **45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	2	3	2	1	1	3	2	3	2	2	3
CO2	3	3	3	2	2	2	3	1	2	2	3	2	2	3	3
CO3	3	3	3	3	3	2	2	2	2	2	3	3	3	3	3
CO4	3	3	3	2	2	3	2	1	2	3	3	2	3	3	3
CO5	2	3	3	2	2	2	3	2	2	3	3	3	3	2	3
CO6	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3
	2.83	2.83	2.83	2.33	2.33	2.5	2.5	1.66	1.83	2.66	2.83	2.66	2.66	2.66	3

Books:

1. D Heam and P M Baker, “Computer Graphics”, Prentice Hall of India (Second Edition), 1995.
2. Woo, Neider, Davis, Shreiner, “OpenGL Programming Guide”, Third Edition, 2000, Pearson Education Asia.
3. A Mukhopadhyay, A. Chattopadhyay, “Introduction to Computer Graphics and Multimedia”, 2nd Edition

Data Analytics Lab (Elective - II Lab) (TIU-UCS-L330D)

Program: B. Tech. in CSE	**Year, Semester:** 3rd Yr., 6th Sem.
Course Title: Data Analytics	**Subject Code:** TIU-UCS-L330D
Contact Hours/Week: 0–0–3(L–T–P)	**Credit:** 1.5

COURSE OBJECTIVE:

1. Understand fundamental statistical analysis methods used in data analytics.
2. Work with Python libraries such as NumPy and Pandas for data handling.
3. Apply machine learning models for data-driven decision-making.
4. Implement ensemble methods for improving prediction accuracy.
5. Perform text mining and sentiment analysis on unstructured data.

COURSE OUTCOME:

Upon completion of the course, the student will be:

CO-1	Understand and apply data preprocessing techniques on real-world datasets.	K2, K3
CO-2	Implement and analyze exploratory data analysis (EDA) techniques to summarize data.	K3,K4
CO-3	Utilize Python libraries such as Pandas, NumPy, and Matplotlib for data manipulation and visualization.	K3,K4
CO-4	Apply machine learning models to structured data and assess their performance.	K3,K4
CO-5	Implement feature engineering and selection techniques for improving model performance.	K3,K4
CO-6	Work with unstructured data such as text and apply text analytics methods	K3, K4

COURSE CONTENT:

MODULE 1:	**Inferential Statistics**	**8 Hours**
Various forms of distribution: Normal, Poisson, Bernoulli. Statistical hypothesis generation and testing, Chi-Square test, Chi-Square test of independence, Z-score, T-test, Z-test vs T-test Analysis of variance (ANOVA), Correlation analysis, Maximum likelihood test.		
MODULE 2:	**Work with Data in Python**	**8 Hours**
Introduction to NumPy: Arrays, mathematical operations, matrix multiplication. Conditional operations, indexing, slicing, shape manipulation. Pandas data structures: Series, DataFrame, Panel. Inserting and exporting data using Pandas.		
MODULE 3:	**Data Analysis by Machine Learning**	**8 Hours**
Supervised, Unsupervised, and Reinforcement learning. Regression, Classification, Clustering. Linear and Logistic Regression, Naive Bayes, Decision Trees. K-Nearest Neighbors, K-Means clustering.		
MODULE 4:	**Pushing Boundaries with Ensemble Models**	**8 Hours**
Basics of Ensemble Methods. Types: Voting and Averaging, Stacking, Bagging, Boosting. Implementation of Random Forest.		
MODULE 5:	**Analyzing Unstructured Data with Text Mining**	**7 Hours**
MODULE 6:	**Data Visualization**	**6 Hours**
Preprocessing text data. Tokenization (word & sentence), POS tagging. Stemming and Lemmatization. Named Entity Recognition (Stanford NER).Sentiment analysis implementation. Introduction to data visualization. Plotting with Matplotlib and Seaborn. Creating interactive visualizations with Plotly. Understanding dashboards and storytelling with data.		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	1	2	2	1	3	2	1	3	2	2
CO2	3	3	2	2	2	1	1	2	1	3	2	1	3	2	2
CO3	3	2	3	2	3	3	2	1	1	3	2	1	3	2	3
CO4	3	3	3	3	3	2	2	1	1	3	3	2	3	3	3
CO5	3	3	3	3	2	3	1	1	1	3	2	1	3	3	3
CO6	2	2	2	2	2	3	3	2	2	3	2	2	3	3	2
	2.83	2.83	2.83	2.33	2.33	2.5	2.5	1.66	1.83	2.66	2.83	2.66	2.66	2.66	3

Books:

Main Reading:

1. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly
2. Samir Madhavan, Mastering Python for Data Science
3. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye, Prentice Hall Inc.

Supplementary Reading

1. Alberto Boschetti, Luca Massaron, Python Data Science Essentials
2. Montgomery, D.C. (2020). *Applied Statistics and Probability for Engineers*. Wiley.
3. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
4. Grus, J. (2019). *Data Science from Scratch: First Principles with Python*. O'Reilly Media.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
6. Jurafsky, D., & Martin, J. (2021). *Speech and Language Processing*. Pearson.

SEMSTER 7

Machine Learning (TIU-UCS-E419C)

Program: B. Tech. in CSE	**Year, Semester:** 4th Yr., 7th Sem.
Course Title: Machine Learning	**Subject Code:** TIU-UCS-E419C
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

Enable the student to:

1. understand the human learning aspects and primitives in learning process by computer
2. analyze the nature of problems solved with machine learning techniques
3. design and implement suitable machine learning technique for a given application

COURSE OUTCOME:

The students will be able to:

CO1:	Explore the underlying principles, mathematical foundations, practical uses, and constraints of current machine learning methods.	K2
CO2:	Recognize the criteria for assessing the effectiveness of the developed model.	K2
CO3:	Investigate and devise contemporary machine learning applications, emphasizing recent advancements and innovative perspectives.	K4
CO4:	Construct the learning model tailored to a specific task.	K3
CO5:	Utilize cutting-edge development frameworks and software libraries to implement	K3
CO6:	Optimize machine learning models by fine-tuning hyperparameters and improving generalization.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION**	**7 Hours**
Definition - Types of Machine Learning - Examples of Machine Learning Problems - Training versus Testing - Characteristics of Machine learning tasks - Predictive and descriptive tasks - Machine learning Models: Geometric Models, Logical Models, Probabilistic Models. Features: Feature types - Feature Construction and Transformation - Feature Selection.		

MODULE 2:	**CLASSIFICATION AND CONCEPT LEARNING**	**7 Hours**
Classification: Binary Classification- Assessing Classification performance - Class probability Estimation - Multiclass Classification - Regression: Assessing performance of Regression - Error measures - Overfitting- Theory of Generalization: Effective number of hypothesis - Bounding the Growth function.		
MODULE 3:	**LINEAR AND PROBABILISTIC MODELS**	**7 Hours**
Least Squares method - Multivariate Linear Regression - Perceptron, Multiple Layer Perceptron - Support Vector Machines - Obtaining probabilities from Linear classifiers - Kernel methods for non-Linearity - Probabilistic models for categorical data – Naïve Bayes Classifier		
MODULE 4:	**DISTANCE BASED MODELS**	**8 Hours**
Distance Based Models: Neighbors and Examples - Nearest Neighbors Classification - Distance based clustering – K-Means Algorithm - K-Medoids Algorithm - Hierarchical clustering - Vector Quantization, Self-Organizing Feature Map - Principal Component Analysis		
MODULE 5:	**RULE BASED AND TREE BASED MODELS**	**8 Hours**
Rule Based Models: Rule learning for subgroup discovery - Association rule mining - Tree Based Models: Decision Trees - Ranking and Probability estimation Trees - Regression trees - Classification and Regression Trees (CART)		
MODULE 6:	**TRENDS IN MACHINE LEARNING**	**8 Hours**
Ensemble Learning, - Bagging and Boosting - Random Forest - Meta learning - Deep Learning - Reinforcement Learning – Applications.		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	1	-	-	-	-	-	-	2	3	2	2
CO2	2	3	2	2	2	-	-	-	-	-	-	2	3	3	2
CO3	2	2	3	2	2	-	-	-	-	-	-	1	3	3	3
CO4	2	2	3	2	3	-	-	-	-	-	-	1	3	3	3
CO5	1	2	2	2	3	-	-	-	-	-	-	1	3	2	3
CO6	2	3	2	3	2	-	-	-	-	-	-	2	3	3	3
	2	2.3	2.3	2.2	2.2							1.5	3	2.67	2.67

Books:

1. P. Flach, "Machine Learning: The art and science of algorithms that make sense of data", Cambridge University Press, 2012, ISBN-10: 1107422221, ISBN-13: 978-1107422223.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Second Edition (Springer Series in Statistics), 2016, ISBN-10: 0387848576, ISBN-13: 978-0387848570.

3. Christopher Bishop, "Pattern Recognition and Machine Learning (Information Science and Statistics)", Springer, 2007.
4. Kevin Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012, ISBN-10: 0262018020, ISBN-13: 978-0262018029
5. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, "Learning from Data", AMLBook Publishers, 2012 ISBN 13: 978-1600490064.
6. Tom Mitchell, "Machine Learning", McGraw-Hill, 1997, ISBN-10: 0071154671, ISBN-13: 978-0071154673.
7. Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques", Chris Ullman, Morgan Kaufmann Publishers,Third Edition, 2011, ISBN 0123814790, ISBN-13 9780123814791

Cloud Computing and Internet of Things (IoT) (TIU-UCS-E419B)

Program: B. Tech CSE	Year, Semester: 4th Year, 7th Sem.
Subject Name: Cloud Computing and Internet of Things (IoT)	Subject Code: TIU-UCS-E419B
Contact Hours/ Weeks: 3-0-0(L-T-P)	Credit: 3

COURSE OBJECTIVE:

Enable the student to:

4. Learn the concepts, models, and architecture of cloud computing.
5. Analyze IaaS, PaaS, and SaaS for different application needs.
6. Work with platforms like AWS, Azure, and Google Cloud for deployment and management.
7. Develop and optimize cloud-based applications for various industries.

COURSE OUTCOME:

The students will be able to:

CO-1:	Explain the fundamental concepts, types, deployment models, and service models of cloud computing along with its benefits.	K2
CO-2:	Classify various cloud service models (IaaS, PaaS, SaaS, IDaaS, CaaS) and illustrate their applications with examples	K3
CO-3:	Describe virtualization techniques, hypervisors, and their role in cloud computing for efficient resource utilization	K2
CO-4:	Analyze the core components, features, and functionalities of major cloud platforms such as Google Cloud, AWS, and Microsoft Azure.	K4
CO-5:	Identify cloud security concerns, security boundaries, encryption techniques, and compliance requirements	K3
CO-6:	Demonstrate cloud management techniques, monitoring strategies, and lifecycle management for cloud services.	K4

COURSE CONTENT:

MODULE 1:	**Basics of Cloud Computing**	**4 Hours**
Defining a Cloud, Cloud Types – NIST Cloud Reference Model, Cloud Cube Model, Deployment Models (Public, Private, Hybrid and Community Clouds), Service Models – Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)Characteristics of Cloud Computing – a shift in paradigm, Benefits and Advantages of Cloud Computing		
MODULE 2:	**Services and Applications by Type**	**8 Hours**
IaaS – Basic Concept, Workload, Partitioning of Virtual Private Server Instances, Pods, Aggregations, Silos, PaaS – Basic Concept, Tools and Development Environment with examples SaaS - Basic Concept and Characteristics, Open SaaS and SOA, examples of SaaS Platform Identity as a Service (IDaaS), Compliance as a Service (CaaS)		
MODULE 3:	**Concepts of Abstraction and Virtualization**	**4 Hours**
Virtualization: Taxonomy of Virtualization Techniques, Hypervisors: Machine Reference Model for Virtualization		
MODULE 4:	**Use of Google Web Services**	**4 Hours**
Discussion of Google Applications Portfolio – Indexed Search, Adwords, Google Analytics, Google Translate, A Brief Discussion on Google Toolkit (including introduction of Google APIs in brief), Major Features of Google App Engine Service		
MODULE 5:	**Use of Amazon Web Services**	**4 Hours**
Amazon Web Service Components and Services: Amazon Elastic Cloud, Amazon Simple Storage System, Amazon Elastic Block Store, Amazon SimpleDB and Relational Database Service		
MODULE 6:	**Use of Microsoft Cloud Services**	**4 Hours**
Windows Azure Platform: Microsoft's Approach, Architecture, and Main Elements, Overview of Windows Azure AppFabric, Content Delivery Network, SQL Azure, and Windows Live Services		
MODULE 7:	**Webmail Services**	**4 Hours**
Cloud Mail Services, including Google Gmail, Windows Live Hotmail, Yahoo Mail		
MODULE 8:	**Cloud-based Storage**	**4 Hours**
Cloud File Systems, including GFS and HDFS		
MODULE 9:	**Cloud Security**	**4 Hours**
Cloud security concerns, security boundary, security service boundary, Overview of security mapping, Security of data: cloud storage access, storage location, tenancy, encryption, auditing, compliance, Identity management (awareness of identity protocol standards)		
MODULE 10:	**Cloud Management**	**5 Hours**
An overview of the features of network management systems and a brief introduction of related products from large cloud vendors, monitoring of an entire cloud computing deployment stack – an overview with mention of some products, Lifecycle management of cloud services (six stages of lifecycle)		

TOTAL LECTURES	45 Hours

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	0	0	2	1	1	1	0	2	0	2	3	2	1
CO2	3	3	2	1	2	1	1	1	1	2	1	2	3	2	1
CO3	3	2	1	1	3	1	1	1	0	2	1	2	3	2	2
CO4	3	3	3	2	3	0	1	0	1	2	2	3	3	2	3
CO5	3	2	1	1	2	2	2	2	0	2	1	2	3	2	1
CO6	3	3	3	2	3	1	1	1	2	3	2	3	3	3	2
	3	2.5	1.7	1.2	2.5	1	1.2	1	0.7	2.17	1.17	2.33	3	2.17	1.67

Books:

8. Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi, "Mastering Cloud Computing: Foundations and Applications Programming", Morgan Kaufmann, 2013, ISBN-10: 0124114547, ISBN-13: 978-0124114548.
9. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", Prentice Hall, 2013, ISBN-10: 0133387526, ISBN-13: 978-0133387520.
10. Arshdeep Bahga and Vijay Madisetti, "Cloud Computing: A Hands-On Approach", CreateSpace Independent Publishing, 2014, ISBN-10: 0996025502, ISBN-13: 978-0996025508.
11. Toby Velte, Anthony Velte, and Robert Elsenpeter, "Cloud Computing: A Practical Approach", McGraw-Hill, 2009, ISBN-10: 0071626948, ISBN-13: 978-0071626941.
12. Michael J. Kavis, "Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)", Wiley, 2014, ISBN-10: 1118617614, ISBN-13: 978-1118617618.

Soft Computing (TIU-UCS-E419D)

Program: B. Tech. in CSE	**Year, Semester: 4th Yr., 7th Sem.**
Course Title: Soft Computing	**Subject Code:** TIU-UCS-E419D
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** Theory–3

COURSE OBJECTIVE:

Enable the student to:

1. Introduce the principles of soft computing techniques, including fuzzy logic, neural networks, and genetic algorithms, and their significance in problem-solving.
2. Explore different soft computing approaches, compare their functionalities, and assess their effectiveness in handling uncertainty, imprecision, and optimization problems.

3. Develop the ability to design and implement soft computing models for real-world applications such as pattern recognition, optimization, and decision-making.
4. Assess the advantages, limitations, and ethical considerations of soft computing methodologies, ensuring their appropriate application in various domains.
5. Explore different soft computing approaches, compare their functionalities, and assess their effectiveness in handling uncertainty, imprecision, and optimization problems.
6. Develop the ability to design and implement soft computing models for real-world applications such as pattern recognition, optimization, and decision-making.
7. Assess the advantages, limitations, and ethical considerations of soft computing methodologies, ensuring their appropriate application in various domains.

COURSE OUTCOME:

The students will be able to:

CO1:	Explain the theoretical foundations of soft computing techniques and their significance.	K1
CO2:	Analyze various soft computing methodologies, including fuzzy logic, neural networks, and genetic algorithms, and assess their effectiveness using theoretical models.	K3
CO3:	Investigate advanced soft computing models and their influence on intelligent problem-solving.	K3
CO4:	Evaluate the performance and limitations of different soft computing techniques in various application scenarios.	K4
CO5:	Identify and mitigate challenges, biases, and constraints in soft computing applications.	K2
CO6:	Implement theoretical principles of soft computing techniques in practical applications, such as pattern recognition, optimization, and decision-making.	K3

COURSE CONTENT:

MODULE 1:	**Introduction**	**7 Hours**
Introduction to soft computing; introduction to fuzzy sets and fuzzy logic systems; introduction to biological and artificial neural network; introduction to Genetic Algorithm.		
MODULE 2:	**Fuzzy Logic**	**10 Hours**
Fuzzy sets and Fuzzy logic systems: Classical Sets and Fuzzy Sets and Fuzzy relations: Operations on Classical sets, properties of classical sets, Fuzzy set operations, properties of fuzzy sets, cardinality, operations, and properties of fuzzy relations. Membership functions: Features of membership functions, standard forms and boundaries, different fuzzification methods. Fuzzy to Crisp conversions: Lambda Cuts for fuzzy sets, fuzzy Relations, Defuzzification methods. Classical Logic and Fuzzy Logic: Classical predicate logic, Fuzzy Logic, Approximate reasoning and Fuzzy Implication		

Fuzzy Rule based Systems: Linguistic Hedges, Fuzzy Rule based system – Aggregation of fuzzy Rules, Fuzzy Inference System-Mamdani Fuzzy Models – Sugeno Fuzzy Models. Applications of Fuzzy Logic: How Fuzzy Logic is applied in Home Appliances, General Fuzzy Logic controllers, Basic Medical, Diagnostic systems and Weather forecasting		
MODULE 3:	**Neural Networks**	**10 Hours**
Introduction to Neural Networks: Advent of Modern Neuroscience, Classical AI and Neural Networks, Biological Neurons andArtificial neural network; model of artificial neuron. Learning Methods: Hebbian, competitive, Boltzman etc., Neural Network models: Perceptron, Adaline and Madaline networks; single layer network; Back-propagation and multi layernetworks. Competitive learning networks: Kohonen self-organizing networks, Hebbian learning; Hopfield Networks. Neuo-Fuzzy modelling: Applications of Neural Networks: Pattern Recognition and classification		
MODULE 4:	**Genetic Algorithms**	**10 Hours**
Simple GA, crossover and mutation, Multi-objective Genetic Algorithm. Applications of Genetic Algorithm: genetic algorithms in search and optimization, GA based clustering Algorithm, Image processing and pattern Recognition		
MODULE 5:	**Other Soft Computing techniques**	**8 Hours**
Simulated Annealing, Tabu search, Ant colony optimization (ACO), Particle Swarm Optimization (PSO).		
TOTAL LECTURES		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2						1		1		2	3		
CO2	3	3	2		2							2	3	2	
CO3	2	3	2	2	2				1	1		2	3		1
CO4	2	2	3		3					1		3	3	2	2
CO5	2	2			2	2	2	2			1	2	2	2	2
CO6	3	2	3	3	3				1	2	2	3	3	3	3
	2.5	2.3	2.3	1	2.3	0.3	0.3	0.5	0.5	1	0.5	2.33	2.83	1.5	1.33

Books:

1. Fuzzy logic with engineering applications, Timothy J. Ross, John Wiley and Sons.
2. S. Rajasekaran and G.A.V.Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms”, PHI
3. Principles of Soft Computing , S N Sivanandam, S. Sumathi, John Wiley & Sons

4. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg
5. Neuro-Fuzzy and Soft computing, Jang, Sun, Mizutani, PHI
6. Neural Networks: A Classroom Approach,1/e by Kumar Satish, TMH,
7. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg, Pearson/PHI
8. A beginners approach to Soft Computing, Samir Roy & Udit Chakraborty, Pearson

Supplementary Reading

1. Fuzzy Sets and Fuzzy Logic: Theory and Applications, George J. Klir and Bo Yuan, Prentice Hall
2. Neural Networks: A Comprehensive Foundation (2nd Edition), Simon Haykin, Prentice Hall.

Machine Learning Lab (TIU-UCS-L419C)

Program: B. Tech. in CSE	**Year, Semester:** 4th Yr., 7th Sem.
Course Title: Machine Learning Lab	**Subject Code:** TIU-UCS-L419C
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE :

Enable the student to:

1. Apply fundamental machine learning techniques using Python.
2. Optimize and fine-tune machine learning models using hyperparameter tuning.
3. Deploy and evaluate machine learning models effectively.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO-1	Illustrate and apply techniques to pre-process data for both supervised and unsupervised learning models using Python.	K3
CO-2	Develop and assess machine learning algorithms for regression, classification, and clustering tasks using Python.	K3
CO-3	Implement and evaluate feature selection, dimensionality reduction, and hyperparameter tuning techniques to enhance model performance.	K4
CO-4	Utilize Python machine learning libraries to build, optimize, and deploy machine learning models.	K4
CO-5	Analyze and implement machine learning algorithms while assessing their performance using appropriate validation techniques.	K4
CO-6	Evaluate machine learning models by applying suitable evaluation metrics and visualization techniques for performance improvement.	K4

COURSE CONTENT:

MODULE 1:	**INTRODUCTION TO PYTHON FOR MACHINE LEARNING**	**6 Hours**
Understanding Python libraries (NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn), Data loading,		

manipulation, and visualization techniques, Data preprocessing: Handling missing values, feature scaling, and encoding		
MODULE 2:	**SUPERVISED LEARNING - REGRESSION & CLASSIFICATION**	**15 Hours**
Implementing Linear Regression and Logistic Regression, Training and evaluating Decision Trees, Random Forests, and Support Vector Machines (SVM), Hyperparameter tuning using Grid SearchCV & Randomized SearchCV		
MODULE 3:	**UNSUPERVISED LEARNING & DIMENSIONALITY REDUCTION**	**6 Hours**
Implementing K-Means Clustering and choosing the optimal K, Feature extraction and Principal Component Analysis (PCA)		
MODULE 4:	**NEURAL NETWORKS**	**6 Hours**
Implementing a Feedforward Neural Network using TensorFlow/Keras, Tuning number of layers, neurons, batch size, and learning rate, Training and testing on MNIST dataset		
MODULE 5:	**NATURAL LANGUAGE PROCESSING**	**6 Hours**
Text tokenization and TF-IDF vectorization, Implementing Naïve Bayes for Sentiment Analysis, Hyperparameter tuning for Naïve Bayes (Laplace smoothing)		
MODULE 6:	**MACHINE LEARNING MODEL DEPLOYMENT**	**6 Hours**
Saving trained ML models using joblib, Creating a Flask API for serving predictions, Testing the deployed model with real-time inputs		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	-	-	2	-	-	-	-	-	-	2	3	3	3
CO2	2	2	2	-	2	-	-	-	-	-	-	2	3	3	3
CO3	3	2	2	2	2	-	-	-	-	-	-	2	3	3	3
CO4	2	-	2	-	3	-	-	-	-	-	-	1	3	3	3
CO5	3	2	2	2	2	-	-	-	-	-	-	2	3	3	3
CO6	2	2	-	-	2	-	-	-	-	-	-	2	3	3	3
	2.3	2	2	2	2.2							1.83	3	3	3

Books:

1. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press.
3. Sebastian Raschka, *Python Machine Learning*, Packt Publishing.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning*, Springer.

5. François Chollet, *Deep Learning with Python*, Manning Publications.

Cloud Computing and IoT Lab (TIU-UCS-L419B)

Program: B. Tech CSE	Year, Semester: 4TH Year, 7th Sem.
Course title: Cloud Computing and IoT Lab	Subject Code: TIU-UCS-L419B
Contact Hours/ Weeks: 0-0-3(L-T-P)	Credit: 1.5

COURSE OBJECTIVE:

1. To provide students with a strong foundation in cloud computing by introducing key concepts, architectures, deployment models (Public, Private, Hybrid, Community), and service models (IaaS, PaaS, SaaS). Students will understand the advantages, challenges, and real-world applications of cloud technologies.
2. To enable students to work hands-on with popular cloud computing platforms such as AWS, Microsoft Azure, and Google Cloud, focusing on deploying, managing, and scaling cloud-based resources. Students will gain experience with cloud storage systems, compute resources, and security fundamentals.
3. To introduce students to IoT protocols (e.g., MQTT, CoAP, HTTP) and connectivity technologies (e.g., Wi-Fi, Bluetooth, Zigbee, LoRa), enabling them to design and implement simple IoT systems using these protocols for communication and integration of sensor data.
4. To help students understand the integration of IoT devices with cloud services, enabling them to send and store IoT data on cloud platforms such as AWS IoT Core and Google IoT Core. This objective aims to give students the ability to leverage cloud computing for real-time IoT data processing and management.

Course outcome:

CO	Course Outcome Description	Bloom's Level
CO1	Understand and apply cloud computing concepts, service models (IaaS, PaaS, SaaS), and deployment models (Public, Private, Hybrid, Community).	K3
CO2	Gain practical experience in working with cloud platforms such as AWS, Azure, and Google Cloud, and utilize cloud resources for deployment and storage.	K3
CO3	Implement and experiment with IoT protocols (MQTT, CoAP, HTTP) and connectivity technologies (Wi-Fi, Bluetooth, Zigbee, LoRa) in building IoT systems.	K4
CO4	Integrate IoT devices with cloud platforms and analyze data using cloud-based IoT services such as AWS IoT Core, Google IoT Core, etc.	K4
CO5	Experiment for real-time processing in IoT systems, and explore AI/ML techniques for IoT data analytics.	K3
CO6	Implement IoT security measures and develop IoT-based smart solutions.	K6

COURSE CONTENT:

Module 1	**Introduction to Cloud Computing**	9 Hours

• Learn cloud computing basics, IaaS, PaaS, SaaS, public and private deployment models, and basic setup of cloud services (AWS/Azure/Google Cloud). • Install and configure VirtualBox, explore hypervisors (KVM, VMware), and understand virtualization concepts in cloud environments.		
Module 2	**Cloud Computing Technologies and Platforms**	9 Hours
• Set up and work with cloud storage services like AWS S3, Google Cloud Storage, and HDFS. Learn about cloud data storage, file management, and retrieval. • Launch EC2 instances, configure basic web applications, manage networking, and set up security measures in the cloud environment.		
Module 3	**Introduction to Internet of Things (IoT)**	12 Hours
• Set up an IoT system with Raspberry Pi/Arduino using MQTT and CoAP protocols for communication. Simulate data transmission and analyze network traffic. • Configure Bluetooth and Zigbee communication between IoT devices. Explore IoT networks and understand data flow and connectivity in low-power networks. • Build a sensor network (e.g., temperature, humidity) using Raspberry Pi or Arduino and integrate it with cloud services for data logging.		
Module 4	**IoT and Cloud Integration & Future Trends**	15 Hours
• Set up an IoT device to send data to AWS IoT Core or Google IoT Core and visualize data on a cloud platform. • Implement an edge computing setup using Raspberry Pi, process IoT data locally, and send processed data to the cloud for storage and analysis. • Apply AI/ML techniques to IoT data for predictive analysis and anomaly detection. Use cloud ML platforms like AWS SageMaker or Google AI. • Develop a complete IoT-based smart solution (e.g., smart home, healthcare monitoring). Integrate IoT devices with the cloud, process data in real time, and implement security and privacy measures. Present the project.		
Total		45Hours

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	0	3	1	1	1	1	2	1	2	3	2	1
CO2	2	2	3	2	3	1	1	1	2	2	2	2	3	3	2
CO3	2	3	3	2	3	1	1	1	2	2	1	2	3	2	3
CO4	3	2	3	2	3	1	1	1	2	2	2	2	3	3	2
CO5	2	2	3	2	2	1	2	1	2	2	2	3	3	3	3
CO6	2	2	3	2	2	2	2	2	2	2	2	3	3	2	3
	2.3	2.2	2.8	1.7	2.7	1.2	1.3	1.2	1.8	2	1.67	2.33	3	2.5	2.33

Books:

1. Cloud Computing: Principles and Paradigms – Rajkumar Buyya, Wiley
2. Mastering Internet of Things – Peter Waher, Packt Publishing
3. Cloud Computing: Theory and Practice – Dan C. Marinescu, Morgan Kaufmann
4. The Internet of Things: Key Applications and Protocols – Olivier Hersent, Wiley
5. Cloud Computing Security – John Rittinghouse, CRC Press

IoT Fundamentals: Networking Technologies, Protocols, and Use Cases – David Hanes, Cisco Press

Soft Computing Lab (TIU-UCS-L419D)

Program: B. Tech. in CSE	**Year, Semester: 4th Yr., 7th Sem.**
Course Title: Soft Computing Lab	**Subject Code:** TIU-UCS-L419D
Contact Hours/Week: 0–0–3	**Credit:** 1.5

COURSE OBJECTIVE:

Enable the student to:

1. Gain fundamental knowledge of soft computing methodologies, including artificial neural networks, fuzzy logic, and genetic algorithms.
2. Develop practical skills in implementing soft computing techniques using programming tools such as Python and MATLAB.
3. Learn how to design, train, and optimize neural networks for classification, regression, and pattern recognition tasks.
4. Develop problem-solving skills by working on mini-projects and case studies related to real-world applications.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO1	Illustrate and apply techniques to pre-process data for both supervised and unsupervised learning models using Python.	K3
CO2	Develop and assess machine learning algorithms for regression, classification, and clustering tasks using Python.	K3
CO3	Implement and evaluate feature selection, dimensionality reduction, and hyperparameter tuning techniques to enhance model performance.	K4
CO4	Utilize Python machine learning libraries to build, optimize, and deploy machine learning models.	K4
CO5	Analyze and implement machine learning algorithms while assessing their performance using appropriate validation techniques.	K4
CO6	Evaluate machine learning models by applying suitable evaluation metrics and	K4

	visualization techniques for performance improvement.	

COURSE CONTENT:

MODULE 1:	**Introduction to Soft Computing Tools**	**4 Hours**
Overview of Soft Computing, Installation and Setup of Python/MATLAB for Soft Computing, Introduction to SciPy, NumPy, and TensorFlow for Soft Computing		
MODULE 2:	**Artificial Neural Networks (ANN)**	**12 Hours**
Implementing Perceptron Learning Algorithm, Training a Multi-Layer Perceptron (MLP) using Backpropagation Implementing a Convolutional Neural Network (CNN) for Image Classification, Training a Recurrent Neural Network (RNN) for Time Series Prediction		
MODULE 3:	**Fuzzy Logic**	**10 Hours**
Fuzzy Set Operations and Membership Functions, Implementing Fuzzy Inference System (Mamdani & Sugeno), Solving a Classification Problem using Fuzzy Logic		
MODULE 4:	**Genetic Algorithms (GA) and Evolutionary Computing**	**8 Hours**
Implementing Genetic Algorithm (GA) for Function Optimization, Solving TSP (Traveling Salesman Problem) using GA		
MODULE 5:	**Hybrid Systems & Applications**	**6 Hours**
Neuro-Fuzzy Systems for Pattern Recognition, Hybrid Soft Computing Approach for Stock Market Prediction		
MODULE 6:	**Project Work**	**5 Hours**
Selecting a real-world problem and solving it using soft computing techniques, Preparing a final project report and presentation		
TOTAL LAB HOURS		**45 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	2	2	3	0	0	1	1	1	0	2	3	2	1
CO2	3	3	3	2	3	0	0	1	1	2	1	2	3	3	2
CO3	2	2	3	2	3	0	0	1	1	2	1	2	3	3	3
CO4	2	2	3	2	3	0	0	0	1	2	1	2	3	3	2
CO5	2	3	2	2	2	0	0	1	1	1	0	2	2	3	2
CO6	2	2	2	2	2	0	0	1	1	1	0	3	2	3	3
	2.3	2.3	2.5	2	2.7	0	0	0.8	1	1.5	0.5	2.17	2.67	2.83	2.17

Books:

- "Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications", S. Rajasekaran, G. A. Vijayalakshmi Pai, Prentice Hall of India (PHI)
- "Soft Computing and Intelligent Systems Design: Theory, Tools and Applications", Fakhreddine O. Karray, Clarence De Silva, Pearson Education
- "Introduction to Soft Computing", Samir Roy, Udit Chakraborty, Pearson Education
- "Soft Computing: Fundamentals and Applications", B. K. Tripathy, J. Anuradha, Oxford University Press

Project-I (TIU-UCS-P495)

Program: B.Tech. in CSE-AI	**Year, Semester:** 4th, 7th.
Course Title: Project-I	**Subject Code:** TIU-UCS-P495
Contact Hours/Week: 0–2–4	**Credit:** 4

COURSE OBJECTIVE:

Enable the student to:

1. Develop students' ability to identify and formulate research problems in various computer Science domains.
2. Enhance students' skills in conducting structured literature reviews to evaluate existing research, technologies, and frameworks.
3. Equip students with analytical skills to identify research gaps and define clear, well-structured research objectives based on technical and theoretical gaps.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO1	Understand research methodologies and identify relevant computer science research problems.	K2
CO2	Conduct a structured literature review using relevant research sources.	K5
CO3	Analyze gaps in current technologies, frameworks, or algorithms.	K5
CO4	Define clear research objectives and justify their significance.	K6
CO5	Develop a research plan with appropriate methodologies and tools.	K4
CO6	Communicate research findings effectively through technical writing and presentations.	K6

COURSE CONTENT:

Module-1:	**RESEARCH PROBLEM IDENTIFICATION**
Introduction to research methodologies in software engineering, AI, data science, cybersecurity,	

and networking. Identifying industry-relevant and academic research gaps. Ethical considerations in computer science research.	
Module-2:	**LITERATURE REVIEW**
Conducting a structured literature review using IEEE, ACM, Springer, etc. Evaluating existing models, architectures, frameworks, and software solutions. Identifying trends, limitations, and emerging technologies in CSE.	
Module-3:	**IDENTIFYING RESEARCH GAPS & DEFINING OBJECTIVES**
Assessing limitations in existing technologies and approaches. Recognizing gaps in performance, security, scalability, or efficiency. Formulating precise research objectives relevant to software, AI, security, or networking.	

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	3	2	2
CO2	2	2	-	2	-	-	-	-	-	-	-	2	3	2	2
CO3	3	2	2	2	-	-	-	-	-	-	-	2	3	3	2
CO4	2	2	2	-	-	-	-	-	-	-	-	2	2	2	2
CO5	2	-	2	2	1	-	-	-	-	-	2	2	2	3	2
CO6	-	-	-	-	-	-	-	-	2	3	-	2	2	2	2
	2.4	2	2	2	1				2	3	2	2	2.5	2.33	2

Career Advancement & Skill Development-VII: Managerial Economics (TIU-UCS-S403)

Program: B. Tech. in CSE-AI	**Year, Semester: 4th** Yr., 7th Sem.
Course Title: Career Advancement & Skill Development-VII: Managerial Economics	**Subject Code:** TIU-UTR-S403
Contact Hours/Week: 2–0–0 (L–T–P)	**Credit:** 2

COURSE OBJECTIVE:

Enable the student to:

1. introduce fundamental concepts of managerial economics, including cost classification, demand-supply analysis, and market structures.
2. develop analytical skills for applying production, cost, and revenue concepts in business decision-making.
3. equip students with knowledge of macroeconomic principles, including GDP, inflation, and fiscal policies, for understanding economic environments.

COURSE OUTCOME:

The students will be able to:

CO1	Define fundamental concepts of managerial economics, cost classification, and firm objectives.	K1
CO2	Explain demand and supply analysis, elasticity, and business forecasting techniques.	K2
CO3	Apply production and cost concepts, including cost-output relationships and returns to scale.	K3
CO4	Analyze revenue types, pricing policies, and the impact of government regulations on pricing.	K3
CO5	Compare different market structures, including perfect and imperfect competition, and oligopolistic strategies.	K4
CO6	Evaluate macroeconomic concepts such as GDP, inflation, fiscal and monetary policies, and balance of payments.	K4

COURSE CONTENT:

Module-1:	**INTRODUCTION**	**5 Hours**
Fundamental Concepts of Managerial Economics, Factors Responsible for Managerial Decision, Cost Concept & Classification, Objectives of the Firm, Correlation Between Productivity and Profitability.		
Module-2:	**DEMAND AND SUPPLY ANALYSIS**	**5 Hours**
Meaning, Types and Determinants, Demand Estimation, Demand Elasticities for Decision Making, Business and Economic Forecasting (Qualitative and Quantitative Methods), Supply Analysis (Meaning, Elasticities, and Determinants), Market Equilibrium.		
Module-3:	**PRODUCTION ECONOMICS**	**5 Hours**
Production and Production Function (Types & Estimation), Cost-Output Relationship, Short-Run and Long-Run Cost Curves, Law of Variable Proportion, Returns to Scale, Economies and Diseconomies of Scale, Economies of Scope, Factor Inputs.		
Module-4:	**REVENUE ANALYSIS AND PRICING POLICIES**	**5 Hours**
Revenue Types, Relationship Between Total Revenue and Price Elasticity of Demand, Pricing Policies and Practices (Objectives, Determinants, Pricing Methods), Government Policies and Pricing.		
Module-5:	**MARKET STRUCTURE**	**5 Hours**
Perfect Competition, Imperfect Competition (Monopoly, Monopolistic, Oligopoly), Oligopolistic Strategy, Cartels, Kinked Demand, Price Leadership, Oligopolistic Rivalry & Theory of Games, Measurement of Economic Concentration, Policy Against Monopoly and Restrictive Trade Practices, Competition Law.		
Module-6:	**INTRODUCTION TO MACROECONOMICS**	**5 Hours**
Circular Flow of Income and Expenditures, Components of National Income and Its Significance, Multiplier Concept, Measuring Gross Domestic Product (GDP), Inflation and Business Cycles,		

Government Fiscal and Monetary Policy, Balance of Payments, Foreign Exchange Markets.	
TOTAL LECTURE	**30 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	-	-	-	-	-	-	-	-	-	-	2	1	-	-
CO2	2	2	-	-	-	-	-	-	-	-	-	2	1	-	-
CO3	3	2	-	-	-	-	-	-	-	-	-	2	2	-	-
CO4	2	2	-	-	-	2	1	2	-	-	-	-	1	-	-
CO5	2	2	-	-	-	-	-	2	-	-	-	-	1	-	-
CO6	2	2	-	-	-	2	3	2	-	-	-	2	1	-	-
	2.166	2				2	2	2				2	1.166		

Books:

1. Mote, Paul and Gupta: Managerial Economics- Concepts and Cases, Tata McGraw Hill,2007
2. Peterson and Lewis: Managerial Economics, 4th Ed., Prentice Hall, 2004
3. Dholakia and Oza: Microeconomics for Management Students, 2nd Edition, Oxford University Press
4. Bhatia and Maheshwari: Economics for Engineers, 3rd Edition, Vikas Publishing House,2018.

SEMSTER 8

Project-II (Final Thesis/Dissertation)
(TIU-UCS-D498)

Program: B.Tech. in CSE-AI	**Year, Semester:** 4th, 8th.
Course Title: Project-II (Final Thesis/Dissertation)	**Subject Code:** TIU-UCS-D498
Contact Hours/Week: 0–4–8	**Credit:** 8

COURSE OBJECTIVE:

Enable the student to:

1. Introduce students to research methodologies and techniques for identifying and formulating research problems in computer science.
2. Equip students with the ability to conduct a structured literature review and critically analyze existing research.
3. Develop students' skills in identifying research gaps and formulating clear, well-defined research objectives aligned with industry and academic needs.

COURSE OUTCOME:

On completion of the course, the student will be able:

CO1	Design and develop a technical solution using programming, algorithms, or frameworks.	K6
CO2	Implement appropriate methodologies for research-driven software or hardware solutions.	K3
CO3	Evaluate and validate project performance using relevant benchmarks.	K5
CO4	Analyze and interpret research findings based on computational experiments.	K4
CO5	Identify potential areas for improvement and propose future research directions.	K4
CO6	Document and present research findings effectively through reports and presentations.	K6

COURSE CONTENT:

Module-1	**SOFTWARE/AI MODEL DEVELOPMENT & THEORETICAL FRAMEWORK**

Developing AI-based solutions, software applications, or network security models. Selecting appropriate datasets, tools, and programming languages. Evaluating model/system performance using metrics like accuracy, latency, security level, etc.	
Module-2	**RESEARCH IMPLEMENTATION & EXPERIMENTATION**
Experimenting with algorithms, software design, or system architecture. Performance benchmarking, debugging, and optimization. Comparing results with existing research to validate improvements.	
Module-3	**CONCLUSION & FUTURE SCOPE**
Summarizing research findings and project outcomes. Discussing industry impact and real-world applications. Identifying future advancements and open research problems.	

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	1	-	-	-	-	-	-	1	3	-	3
CO2	2	3	2	-	1	-	-	-	-	-	-	1	3	2	3
CO3	2	2	-	3	1	-	-	-	-	-	-	2	3	-	2
CO4	2	3	-	3	-	-	-	-	-	1	-	2	2	-	2
CO5	2	2	-	-	-	-	-	-	-	-	-	2	2	2	2
CO6	-	-	-	-	-	-	-	-	3	3	2	1	2	-	-
	2	2.4	2.5	3	1				3	2	2	1.5	2.5	2	2.4

Career Advancement & Skill Development-VII: Values and Ethics (TIU-UMG-S400)

Program: B. Tech. in CSE-AI	**Year, Semester: 4th** Yr., 8th Sem.
Course Title: Career Advancement & Skill Development-VIII: Values and Ethics	**Subject Code:** TIU-UMG-S400
Contact Hours/Week: 2–0–0 (L–T–P)	**Credit:** Sessional–2

COURSE OBJECTIVE:

Enable the student to:

1. prepare students for job interviews by understanding recruiter expectations and effective communication.
2. develop skills in resume writing, cover letter drafting, and SOP preparation for career opportunities.
3. familiarize students with corporate expectations, workplace etiquette, and professional growth strategies.

COURSE OUTCOME:

The students will be able to:

CO1	Describe the interview process and the recruiter's perspective on candidate evaluation.	K1
CO2	Identify various job roles and the associated skills required in the Electronics and Communication Engineering (ECE) domain.	K1
CO3	Apply resume writing, cover letter drafting, and statement of purpose (SOP) preparation techniques.	K3
CO4	Analyze common interview and written test questions for specific job roles and develop response strategies.	K4
CO5	Explain corporate expectations from newly joined employees and the importance of workplace etiquette.	K2
CO6	Develop strategies to adapt to corporate culture, manage time effectively, and build professional networks.	K3

COURSE CONTENT :

Module-1:	**INTERVIEW PROCESS AND APPROACH**	**8 Hours**
Interview Process from Recruiter's Perspective, Types of Interviews (HR, Technical, Behavioral, Case-Based), Job Description Analysis, Role-Specific Interview Preparation, Effective Communication and Body Language, Mock Interviews and Feedback.		
Module-2:	**JOB ROLES AND INTERVIEW PREPARATION**	**6 Hours**
Job Roles in Electronics and Communication Engineering (ECE), Responsibilities and Required Skills, Common Interview and Written Test Questions, Technical and Aptitude-Based Question Strategies, Industry Expectations from Fresh Graduates.		
Module-3:	**RESUME, COVER LETTER, AND SOP WRITING**	**8 Hours**
Resume Structure and Key Components, Tailoring Resumes for Different Roles, Common Resume Mistakes, Cover Letter Writing (Format and Customization), Statement of Purpose (SOP) Writing (Purpose and Structure), Resume and SOP Review Sessions.		
Module-4:	**CORPORATE EXPECTATIONS AND SUCCESS STRATEGIES**	**8 Hours**
Employer Expectations from New Employees, Workplace Etiquette and Professionalism, Adapting to Corporate Culture, Time Management and Productivity, Networking and Professional Relationship Building, Overcoming Common Workplace Challenges.		
	TOTAL LECTURE	**30 Hours**

CO-PO MATRIX:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	-	-	-	-	-	1	-	-	2	3	-	1	1	-	-
CO2	2	2	-	-	-	-	-	-	-	1	-	2	2	-	-
CO3	-	-	-	-	-	-	-	-	2	3	1	1	1	-	-

CO4	1	2	-	-	-	-	-	-	1	2	-	2	2	-	-
CO5	-	-	-	-	-	2	-	1	2	2	-	2	-	-	-
CO6	-	-	-	-	-	-	-	-	3	2	2	2	-	-	-
	1.5	2				1.5		1	2	2.5	1.5	2	2.5		

Books:

1. Bolles, R. N. (2018). What color is your parachute? A practical manual for job-hunters and career-changers. Ten Speed Press.
2. Yate, M. (2017). Knock 'em dead: The ultimate job search guide. Adams Media.
3. Guffey, M. E., & Loewy, D. (2016). Essentials of business communication (10th ed.). Cengage Learning.
4. Carnegie, D. (2011). How to win friends and influence people in the digital age. Simon & Schuster.
5. Weiss, J. (2014). Welcome to the real world: Finding your place, perfecting your work, and turning your job into your dream career. Hachette Books.

Natural Language Processing (NLP) and its applications (TIU-UCS- E418)

Program: B. Tech. in CSE	Year, Semester: 4th Yr., 8th Sem.
Course Title: Natural Language Processing (NLP) and its applications	Subject Code: TIU-UCS-E418
Contact Hours/Week: 3–0–0 (L–T–P)	Credit: 3

COURSE OBJECTIVE:

1. Introducing cutting-edge systems and trends in natural language processing to the students.
2. Make sure they comprehend the language's morphology, syntax, semantics, and pragmatic notions and can provide the necessary examples to support the aforementioned ideas.
3. Teach them the importance of pragmatics in interpreting natural language.
4. Give students the tools they need to explain a natural language processing application and to demonstrate syntactic, semantic, and pragmatic processing.

COURSE OUTCOME:

The students will be able to:

CO1:	Explain the fundamental concepts of Natural Language Processing (NLP), including syntax, semantics, and pragmatics.	K2

CO2:	Apply various text preprocessing techniques such as tokenization, stemming, and lemmatization to prepare data for NLP tasks.	K3
CO3:	Analyze different NLP models like N-grams, Hidden Markov Models (HMM), and neural networks to solve language-based problems.	K3
CO4:	Evaluate the performance of NLP algorithms using appropriate metrics (e.g., accuracy, precision, recall, and F1 score).	K4
CO5:	Design and implement NLP applications such as sentiment analysis, machine translation, and chatbots using modern frameworks (e.g., NLTK, Spacy, or TensorFlow).	K3
CO6:	Critically assess the ethical considerations and biases in NLP models and their real-world impact.	K3

COURSE CONTENT:

MODULE 1:	**Introduction to NLP**	**10 Hours**
Natural language processing issues and strategies. Tools of NLP, Linguistic organization of NLP, NLP as an Application domain. Word Classes: Regular Expressions: Chomsky hierarchy, CFG and different parsing techniques, Morphology: Inflectional, derivational, parsing and parsing with FST, Combinational Rules, Joint and conditional probability. Probabilistic Language modeling and its Applications.		
MODULE 2:	**Language Modeling and Naïve Bayes**	**14 Hours**
Markov models, N- grams. Estimating the probability of a word and smoothing. Counting words in Corpora, simple N-grams, smoothing (Add One, Written-Bell, Good-Turing). Part of Speech Tagging and Hidden Markov Models: Part of Speech tagging, Indian Language on focus Morphology Analysis, Accuracy Measure and Probability, HMM, Viterbi algorithm for finding most likely HMM Path. HMM tagging, transformation based tagging. Probabilistic Context Free Grammars: Weighted context free grammars.		
MODULE 3:	**Semantics**	**12 Hours**
Representing Meaning: Unambiguous representation, canonical form, expressiveness, meaning structure of language Semantic Analysis: NLP and IR, How NLP has used IR Towards Latent Semantic. Lexical Semantics: Lexemes(synonymy, hyponymy etc), WordNet, metonymy and their computational approaches Supervised and Unsupervised methods Word Sense Disambiguation: Selectional restriction based, machine learning based and dictionary based approaches.		
MODULE 4:	**Pragmatics**	**9 Hours**

Information Theory: Entropy, Cross-entropy, information gain. Reference resolution and phenomena, syntactic and semantic constraints. Pronoun resolution algorithm, text coherence, and discourse structure Natural Language Generation: Introduction to language generation, architecture, discourse planning (text schemata, rhetorical relations). Resource Constrained WSD, Parsing Algorithms, Parsing Ambiguous Sentences, Probabilistic Parsing Algorithms.

TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	1	-	-	-	-	-	-	2	3	2	2
CO2	2	2	-	2	3	-	-	-	-	-	-	1	3	3	3
CO3	3	3	2	2	2	-	-	-	-	-	-	1	3	3	3
CO4	2	3	1	2	2	-	-	-	-	-	-	1	3	3	3
CO5	2	2	2	2	3	-	-	-	-	-	-	1	3	3	3
CO6	2	2	-	-	-	2	-	2	-	1	-	1	3	2	2
	2.33	2.33	1.67	2	2.2	2		2		1		1.17	3	2.67	2.67

Books:

1. D. Jurafsky & J. H. Martin – "Speech and Language Processing – An introduction to Language processing, Computational Linguistics, and Speech Recognition", Pearson Education
2. Allen, James. 1995. – "Natural Language Understanding". Benjamin/Cummings, 2ed. Bharathi, A., Vineet Chaitanya and Rajeev Sangal. 1995.
3. Natural Language Processing- "A Pananian Perspective". Prentice Hall India, Eastern Economy Edition. 3. Eugene Cherniak: "Statistical Language Learning", MIT Press, 1993.
4. Manning, Christopher and Heinrich Schutze. 1999. "Foundations of Statistical Natural Language Processing". MIT Press.
5. Cognitively Inspired Natural Language Processing Abhijit Mishra, Pushpak Bhattacharyya Springer.

Data Warehousing and Data Mining (TIU-UCS-E416)

Program: B. Tech. in CSE	**Year, Semester:** 4th Yr., 8th Sem.
Course Title : Data Warehousing and Data Mining	**Subject Code:**TIU-UCS-E416

Contact Hours/Week:3–0–0(L–T–P)	**Credit:**3

COURSE OBJECTIVE:

1. Introducing cutting-edge systems and trends in Data Mining to the students.
2. Make sure they comprehend the Data analysis to provide the necessary examples to support the aforementioned ideas.
3. Teach them the importance of Data base analysis.
4. Give students the tools they need to explain Data warehousing and Data Mining and it's application.

COURSE OUTCOME:

The students will be able to:

CO-1:	To understand the basic principles, concepts and applications of data warehousing and data mining	K2
CO-2:	To introduce the task of data mining as an important phase of knowledge recovery process.	K3
CO-3:	Ability to do Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment.	K4
CO-4:	Have a good knowledge of the fundamental concepts that provide the foundation of data mining.	K5
CO-5:	Design a data warehouse or data mart to present information needed by management in a form that issuable for management client.	K6
CO-6:	Getting Knowledge from data , Find pattern and finally give some prediction.	K5

COURSE-CONTENT:

MODULE1:	Overview of Data warehousing	**3Hours**
Overview of Data warehousing, Strategic information and the need for Data warehousing, Defining a Data warehouse, Evolution of Data warehousing, Data warehousing and Business Intelligence.		

MODULE2:	The Building Blocks of Data warehouse	**10Hours**
Defining features–Subject-oriented data, Integrated data, Time-variant data. Nonvolatile data, Data granularity Data warehouses and Data marts Architectural Types–Centralized, Independent data marts, Federated, Hub-and-Spoke, Data mart bus Overview of components-Source Data, Data Staging, Data Storage, Information Delivery, Metadata, and Management and Control components. The STAR schema – illustration, Dimension Table, Fact Table, Factless Fact Table, Data granularity, STAR schema keys – Primary, Surrogate, and Foreign, Advantages of the STAR schema, STAR schema examples. Overview of ETL, Requirements of ETL		
MODULE3:	Business Requirements and Data warehouse	**9Hours**
Business Requirements and Data warehouse: Dimensional nature of Business data and Dimensional Analysis, Dimension hierarchies and categories, Key Business. Metrics (Facts), Requirement Gathering methods and Requirements Definition Document (contents). Dimensional nature of Business data and Dimensional Analysis, Dimension hierarchies and categories, Key Business Metrics (Facts), Requirement Gathering methods and Requirements Definition Document (contents) Business Requirements and Data Design–Structure for Business Dimensions and Key Measurements, Levels of detail Business Requirements and the Architecture plan Business Requirements and Data Storage Specifications Business Requirements and Information Delivery Strategy.		
MODULE4:	Architectural components	**10Hours**
Concepts of Data warehouse architecture–Definition and architecture in the areas of Data acquisition, Data storage, and Information delivery Distinguishing characteristics Different objectives and scope. Data content, Complex analysis for faster response, Flexible and Dynamic, Metadata-driven etc.Architectural Framework–supporting flow of data, and the Management and Control module Technical architecture–Data acquisition, Data storage, and Information delivery Overview of the components of Architectural. Metadata types by functional areas–Data acquisition, Data storage, and Information delivery Business Metadata–overview of content and examples Technical Metadata–overview of content Metadata Requirements, Sources of Metadata, Metadata management–challenges, Metadata Repository, Metadata integration and standards		
MODULE5:	Data Mining Analysis Technique	**13Hours**

Information from Data warehouse versus Operational systems, Users of information–their needs and how to provide information, Information delivery–queries, reports, analysis, and applications, Information delivery tools–Desktop environment, Methodology and criteria for tool selection, Information delivery framework, Business Activity Monitoring, Dashboards and Scorecards, OLAP in Data warehouse Overall concept of Online Analytical Processing (OLAP), OLAP definitions and rules, OLAP characteristics Major features and functions of OLAP–General features, Dimensional analysis, Hypercubes, Drill Down and Roll Up, Slice and Dice, Rotation, Uses and Benefits Data Warehouse and the web Web-enabled Data Warehouse–adapting data warehouse for the web, Web-based information delivery–Browser technology for datawarehouse and Security issues, OLAP and Web–Enterprise OLAP, Web-OLAP approaches, OLAP Engine design. Data Mining Overview of Data mining–Definition, Knowledge Discovery Process (Relationships, Patterns, Phases of the process), OLAP versus Datamining, Some aspects of Data mining–Association rules, Outlier analysis, Predictive analytics etc), Concepts of Data mining in a Data warehouse environment, Major Data Mining techniques Cluster Detection, Decision Trees, Memory-based Reasoning, Link Analysis, Neural Networks, Genetic Algorithms etc, Data Mining Applications in industry Benefits of Data mining, Discussion on applications in Customer Relationship Management (CRM), Retail, Telecommunication, Biotechnology, Banking and Finance etc

TOTAL LECTURES	**45 Hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	1	-	-	-	-	-	-	2	3	3	3
CO2	2	2	2	2	2	-	-	-	-	-	-	1	3	3	3
CO3	3	3	1	2	2	-	-	-	-	-	-	1	3	3	3
CO4	2	2	-	2	3	-	-	-	-	-	-	1	3	3	3
CO5	1	1	-	-	1	-	-	-	2	3	-	1	2	2	2
CO6	2	-	-	-	-	2	-	2	-	2	-	1	2	2	2
	2.17	2	1.5	2	1.8	2		2	2	2.5		1.17	2.67	2.67	2.67

Textbooks:

1. Data Mining Technology, Third Edition by Arun K Pujari, Universities Press, India

2. Data Warehousing Fundamentals for IT Professionals, Second Edition by Paulraj Ponniah, Wiley India

3. Alex Berson, Stephen J. Smith, “Data Warehousing Data Mining & OLAP”, Tata McGrow- Hill References

References:

1. Data Warehousing, Data Mining, & OLAP – Second Edition by Alex Berson and Stephen J. Smith, Tata McGraw Hill

2. Data warehouse Toolkit by Ralph Kimball, Wiley India

Gajendra Sharma,“Data Mining Data Warehousing and OLAP”, S.K.KATARIA & SONS

Distributed Operating System (TIU-UCS-E422)

Program: B. Tech. in CSE	**Year Semester:** 4th Yr.,8th Sem.
Course Title: Distributed Operating System	**Subject Code:** TIU-UCS-E422
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit:** 3

COURSE OBJECTIVE:

Enable the student to:

1. Understand the Fundamentals of Distributed Systems.
2. Design and Implement Distributed Algorithms.
3. Explore Distributed File Systems and Storage.
4. Evaluate performance and fault tolerance of distributed systems.

COURSE OUTCOME:

The students will be able to:

CO1	Focuses on foundational knowledge, addressing key concepts of distributed systems and transparency issues.	K2
CO 2	Introduces distributed paradigms like message passing, client-server communication, and RPC, applied to various programming environments.	K2
CO 3	Emphasizes communication mechanisms such as socket programming and secure communication protocols.	K3
CO4	Focuses on process models and synchronization techniques, enabling students to apply and analyze complex synchronization algorithms.	K4
CO5	Focuses on consistency, replication, and distribution protocols, ensuring students can design and develop fault-tolerant systems.	K4
CO6	Addresses fault tolerance in distributed systems, enabling students to design reliable systems and understand recovery techniques.	K3

COURSE CONTENT:

MODULE 1:	**Introduction to Distributed systems**	**6 Hours**
Review of Networks, Operating Systems, Concurrent Programming, and Characteristics & Properties of Distributed Systems – Taxonomy - Design goals – Transparency Issues		
MODULE 2:	**Distributed Computing Paradigms**	**8 Hours**

Basic Message Passing Model – The Client Server, Message Passing, RPC basics, RPC implementation, RPC communication and issues, Remote Procedure Call Model – RPC in conventional languages and in Java - The Distributed Objects – The Collaborative Application		
MODULE 3:	**Inter process communication mechanisms**	**7 Hours**
Communication in Distributed Systems, Socket Programming -Client Server examples, I/O Multiplexing, Inetd Super Server – Secure Sockets – The SSL & the Java Secure Socket Extension		
MODULE 4:	**Process models in distributed systems**	**8 Hours**
Processes, Threads - Code Migration Software Agents – CSP Distributed Processes - Naming with Mobile Entities - Unreferenced Objects Synchronization: Clock Synchronization – Logical clocks – Election Algorithms – Distributed Mutual Exclusion.		
MODULE 5:	**Consistency and Replication**	**4 Hours**
Motivation, Object Replication, Consistency Models, Distribution Protocols , Consistency Protocols		
MODULE 6:	**Fault Tolerance**	**4 Hours**
Failure Models – Process Resilience – Reliable Client Server and Group Communications – Distributed Commit Protocols – Check-pointing and Recovery - Distributed Databases - Distributed Transactions.		
MODULE 7:	**Distributed File System**	**8 Hours**
File system, DFS- definition, Characteristics, Goals, SUN NFS-NFS Architecture, NFS Implementation, Protocols, The CODA file system-Design Overview, An Example, Design Rational, Implementation, The GOOGLE file system-Definition, Architectures, GFS Architecture		
Total		45 hours

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	0	0	1	1	1	0	2	3	2	1

CO2	3	3	2	2	3	1	0	1	1	2	1	2	3	3	2
CO3	2	3	2	2	3	0	0	1	1	2	1	2	3	3	3
CO4	3	3	3	3	2	0	0	1	2	2	1	3	3	3	3
CO5	3	3	3	2	2	0	0	1	2	2	2	3	3	3	3
CO6	2	2	2	2	2	0	1	2	2	2	2	3	3	3	3
	2.67	2.67	2.17	2	2.33	0.17	0.17	1.17	1.5	1.83	1.17	2.5	3	2.83	2.5

Books:

1. Andrew S. Tannenbaum and Maarten Van Steen, Distributed Systems: Principles and Paradigms, Pearson
2. George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, Distributed Systems: Concepts and Design, Addison Wesley

Reference Books:

1. P. K. Sinha, Distributed Operating Systems: Concepts and Design, IEEE press
2. Chandda Roy, Distributed Database Systems, Pearson Education India
3. M. Singhal and N. G. Shivaratri, Advanced Concepts in Operating Systems,, McGraw-Hill

Bioinformatics (TIU-UCS-E420)

Program: B. Tech. in CSE	**Year, Semester**: 4th Year, 8th Semester
Course Title: Bioinformatics	**Subject Code**: TIU-UCS-E420
Contact Hours/Week: 3–0–0 (L–T–P)	**Credit**: Theory – 3

COURSE OBJECTIVE

1. Gain a solid understanding of the core principles and concepts in bioinformatics.
2. Develop skills in DNA, RNA, and protein sequence alignment, similarity searching.
3. Understand how to utilize and navigate biological databases, such as GenBank, PDB, and UniProt, for retrieving biological data.
4. Learn methods for analyzing high-throughput data, including gene expression, microarray, and RNA-Seq data.

COURSE OUTCOME

CO1	Understand the genesis of Bioinformatics, comparison with its allied disciplines, theoretical and computational models and its significance in biological data analysis.	K2
CO2	Explain nucleic acid and protein sequence databases, structural databases, literature databases, genome and organism-specific	K4

	databases.	
CO3	Describe retrieval tools of biological data, database similarity searching, biological file formats.	K3
CO4	Analysis and development of models for better interpretation of biological data to extract knowledge.	K3
CO5	Apply machine learning and statistical techniques for biological data analysis	K3
CO6	Develop bioinformatics applications using computational tools and programming.	K2

COURSE CONTENT

MODULE 1:	**Introduction**	**8 Hours**
Introduction to bioinformatics; Bioinformatics Applications; Central Dogma of Molecular Biology; Genome projects; Sequence analysis, Homology and Analogy.		
MODULE 2:	**Biological Resources**	**7 Hours**
Protein Information Resources**:** Introduction; Biological databases; Primary Sequence Databases; Composite Protein Sequence Databases; Secondary Databases; Composite protein pattern databases; Structure classification databases Genome Information Resources: Introduction; DNA sequence databases; Specialized Genomic Resources		
MODULE 3:	**DNA sequence analysis**	**6 Hours**
Gene structure and DNA sequence; Features of DNA sequence analysis; Issues in interpretation of EST searches; Different approaches to EST analysis; Effects of EST data on DNA databases.		
MODULE 4:	**Pairwise Sequence Alignment**	**7 Hours**
Database searching; Alphabet and Complexity; Algorithms and Programs; Comparing two sequences; Identity and Similarity; Local and global similarity; Global alignment: the Needleman and Wunsch algorithm; Local alignment: the Smith-Waterman algorithm; Dynamic Programming; Pairwise database searching; Basic Local Alignment Search Tool (BLAST).		
MODULE 5:	**Multiple Sequence alignment**	**7 Hours**
Goal of Multiple Sequence Alignment (MSA); Purpose of MSA; Dynamic programming solution for multiple alignment; Methods of alignment.		
MODULE 6:	**Protein Secondary Structure Predictions**	**5 Hours**

Structure of protein; Different level of protein structure; Basics of machine learning; Methods for predicting secondary structure: Chou-Fasman method, Garnier-Osguthorpe-Robson method, Neural Network based method.		
MODULE 7:	**Biomedical Text Mining**	**5 Hours**
Named entity recognition; Document classification and clustering; Relationship discovery; Information extraction; Information retrieval and question answering; Applications of biomedical text mining.		
TOTAL LECTURES		**45 hours**

CO-PO MATRIX:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	1	2	2	-
CO2	2	1	-	-	-	-	-	-	-	-	-	-	2	-	-
CO3	2	2	-	-	1	-	-	-	-	-	-	1	2	-	-
CO4	3	2	-	2	-	-	-	-	-	-	-	2	2	2	1
CO5	2	3	-	2	1	-	-	-	-	-	-	2	2	3	2
CO6	2	2	2	-	2	-	-	-	1	-	-	2	3	2	3
	2.33	2	2	2	1.33				1			1.6	2.17	2.25	2

Books:

1. T K Attwood, D J Parry-Smith, Samiron Phukan; Introduction to bioinformatics, Pearson
2. S. C. Rastogi, P. Rastogi, N. Mendiratta; Bioinformatics Methods And Applications: Genomics Proteomics And Drug Discovery, PHI.
3. Bryan Bergeron, Bioinformatics Computing, Pearson
4. S. Harisha, Fundamentals of Bioinformatics, I.K International