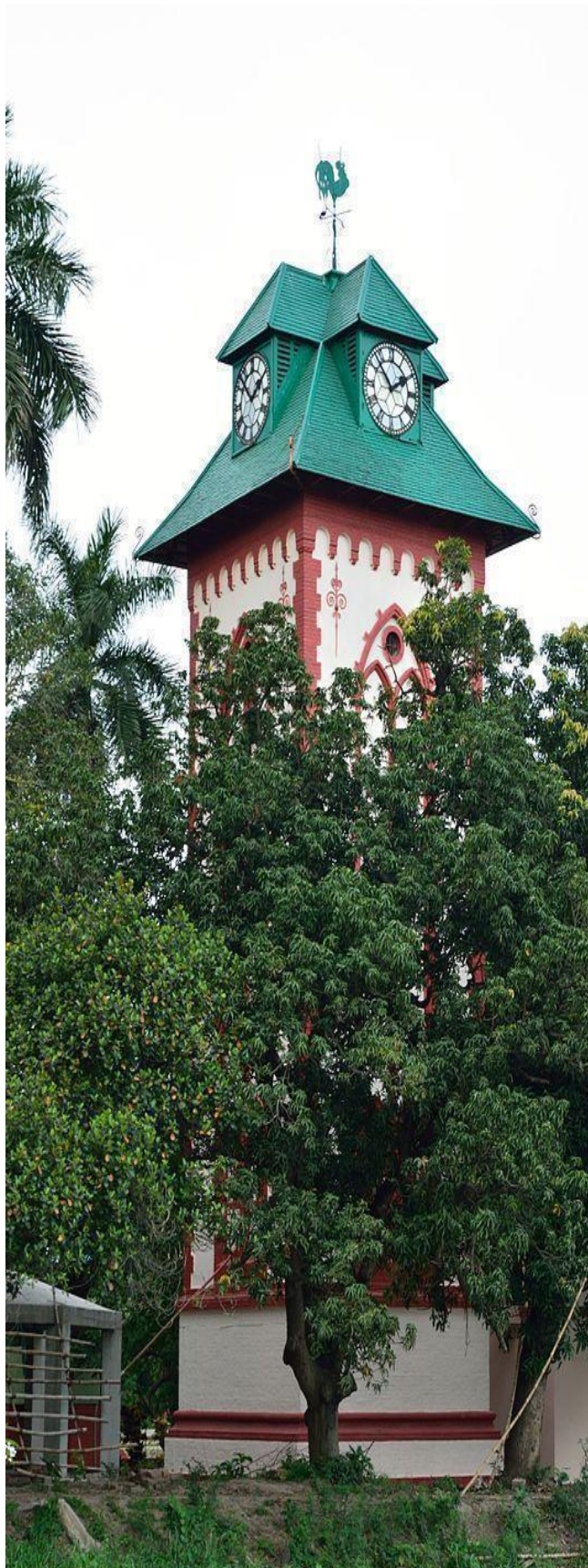




**Indian Institute of Engineering
Science and Technology (IIEST),
Shibpur**

Botanic Garden, Howrah

**Department of
Information Technology**

B.Tech Programme
Course Structure and Syllabus

(Effective from 2025-26 admitting batch onwards)

Table of Contents

Table of Contents

Course Structure

B.Tech in Information Technology

1st Semester Courses Syllabi

2nd Semester Courses Syllabi

3rd Semester Course Syllabi

4th Semester Course Syllabi

5th Semester Course Syllabi

6th Semester Course Syllabi

7th Semester Course Syllabi

8th Semester Course Syllabi

COURSE STRUCTURE FOR B. TECH. (IT)

in Information Technology

COURSE STRUCTURE FOR B. TECH. IN INFORMATION TECHNOLOGY									
First Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	BSC	Engineering Mathematics – I	MA1101N	3	1	0	4	4	100
2	BSC	Engineering Chemistry	CH1101N	3	0	0	3	3	100
3	ESC	Introduction to Computing	CS1101N	3	0	0	3	3	100
4	ESC	Basic Electronics	ET1101N	3	0	0	3	3	100
5	VAC	Well-being and Happiness	HU1101N	2	0	0	2	2	50
6	HSC	Professional Communication in English	HU1102N	2	1	0	3	3	100
		Theory Sub-total		16	2	0	18	18	550
7	ESC	Workshop	WS1171N	0	0	3	2	3	50
8	BSC	Engineering Chemistry Lab	CH1171N	0	0	3	2	3	50
9	ESC	Computer Programming Practice Lab	CS1171N	0	0	3	2	3	50
10		NSS/NCC/PT/Yoga					R*		
		Practical Sub-total		0	0	9	6	9	150
		Second Semester Total		16	2	9	24	27	700

Second Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	BSC	Engineering Mathematics – II	MA1201N	3	1	0	4	4	100
2	BSC	Engineering Physics	PH1201N	3	0	0	3	3	100
3	ESC	Basic Electrical Engineering	EE1201N	3	0	0	3	3	100
4	ESC	Introduction to AI and ML	CS1202N	3	0	0	3	3	100
5	VAC	Energy, Environment and Climate Change	CE1201N	2	0	0	2	2	50
6	PC	Discrete Mathematics and Graph Theory	IT1201N	3	1	0	4	4	100
		Theory Sub-total		17	2	0	19	19	550
7	ESC	Engineering Graphics	AM1271N	0	0	3	2	3	50
8	BSC	Physics Lab	PH1271N	0	0	3	2	3	50
9	ESC	Basic Electrical Engineering Lab	EE1271N	0	0	3	2	3	50
10	PC	Introduction to Departmental Labs (Fundamentals of Programming)	IT1271N	0	0	3	2	3	50
11		NSS/NCC/PT/Yoga					R*		
		Practical Sub-total		0	0	12	8	12	200
		Second Semester Total		17	2	12	27	31	750

-

<u>Third Semester</u>									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	BSC	Mathematics - III	MA2101N	3	0	0	3	3	100
2	PC	Data Structure and Algorithms	IT2101N	3	1	0	4	4	100
3	PC	Digital Logic and Circuit Design	IT2102N	3	1	0	4	4	100
4	PC	Signals and Systems	IT2103N	3	0	0	3	3	100
5	PC	Formal Languages and Automata Theory	IT2104N	3	0	0	3	3	100
		Theory Sub-total		15	2	0	17	17	500
6	PC	Data Structure and Algorithms Lab	IT2171N	0	0	3	2	3	50
7	PC	Digital Logic and Circuit Design Lab	IT2172N	0	0	3	2	3	50
8	PC	Object Oriented Programming Lab	IT2173N	0	0	3	2	3	50
9	PC	Mini Project	IT2191N	0	0	3	2	3	50
		Practical Sub-total		0	0	12	8	12	200
		Third Semester Total		15	2	12	25	29	700

<i>Fourth Semester</i>									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	PC	Foundations of Programming Languages and Translation	IT2201N	3	0	0	3	3	100
2	PC	Communication Systems	IT2202N	3	0	0	3	3	100
3	PC	Computer Organization and Architecture	IT2203N	3	1	0	4	4	100
4	PC	Design and Analysis of Algorithms	IT2204N	3	1	0	4	4	100
5	OE	<i>Refer to the list below</i>	IT226xN	3	0	0	3	3	100
		Theory Sub-total		15	2	0	17	17	500
6	PC	Foundations of Programming Languages and Translation Lab	IT2271N	0	0	3	2	3	50
7	PC	Communication Systems Lab	IT2272N	0	0	3	2	3	50
8	PC	Computer Organization and Architecture Lab	IT2273N	0	0	3	2	3	50
9	PC	Algorithms Lab	IT2274N	0	0	3	2	3	50
		Practical Sub-total		0	0	12	8	12	200
		Fourth Semester Total		15	2	12	25	29	700

OE for Fourth Semester	Course Code
1. Multimedia Systems	IT2261N

<i>Fifth Semester</i>									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	PC	Microprocessor and Microcontroller	IT3101N	3	0	0	3	3	100
2	PC	Database Management Systems	IT3102N	3	1	0	4	4	100
3	PC	Operating Systems	IT3103N	3	1	0	4	4	100
4	PSE	<i>Refer to the list below</i>	IT312xN	3	0	0	3	3	100
5	OE	<i>Refer to the list below</i>	IT316xN	3	0	0	3	3	100
		Theory Sub-total		15	2	0	17	17	500
6	PC	Microprocessor and Microcontroller Lab	IT3171N	0	0	3	2	3	50
7	PC	Database Management Systems Lab	IT3172N	0	0	3	2	3	50
8	PC	Operating Systems Lab	IT3173N	0	0	3	2	3	50
		Practical Sub-total		0	0	9	6	9	150
		Fifth Semester Total		15	2	9	23	26	650

PSE for Fifth Semester	Course Code	OE for Fifth Semester	Course Code
1. Software Engineering	IT3121N	1. Image Processing	IT3161N
2. Mobile Communications	IT3122N		
3. Information and Coding Theory	IT3123N		
4. Soft Computing Techniques	IT3124N		
5. Graph Algorithms	IT3125N		
6. Digital Signal Processing	IT3126N		
7. Design Practices	IT3127N		
8. Computer Graphics	IT3128N		
9. Compiler Design	IT3129N		
10. Fundamentals of Information Retrieval	IT3130N		

<i>Sixth Semester</i>									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	PC	Computer Networks	IT3201N	3	1	0	4	4	100
2	PC	Machine Learning	IT3202N	3	1	0	4	4	100
3	PSE	<i>Refer to the list below</i>	IT322xN	3	0	0	3	3	100
4	VAC	Sociology & Professional Ethics		3	0	0	3	3	100
5	OE	<i>Refer to the list below</i>	IT326xN	3	0	0	3	3	100
		Theory Sub-total		15	2	0	17	17	500
6	PC	Computer Networks Lab	IT3271N	0	0	3	2	3	50
7	PC	Machine Learning Lab	IT3272N	0	0	3	2	3	50
		Practical Sub-total		0	0	6	4	6	100
		Sixth Semester Total		15	2	6	21	23	600

PSE for Sixth Semester		Course Code	OE for Sixth Semester	Course Code
1.	Artificial Intelligence	IT3221N	1. Internet of Things	IT3261N
2.	Real Time Systems	IT3222N		
3.	Embedded Systems and IoT	IT3223N		
4.	High Performance Computer Architecture	IT3224N		
5.	Computational Geometry	IT3225N		
6.	Wireless Networks	IT3226N		
7.	Computer Vision	IT3227N		
8.	Natural Language Processing	IT3228N		

Seventh Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	PC	Information and Systems Security	IT4101N	3	0	0	3	3	100
2	PSE	<i>Refer to the list below</i>	IT412xN	3	0	0	3	3	100
3	HSC	Finance Economics and Management for Engineers		3	0	0	3	3	100
4	OE	<i>Refer to the list below</i>	IT416xN	3	0	0	3	3	100
		Theory Sub-total		12	0	0	12	12	400
5	P	Project - I	IT4191N				4		200
6	PC	Information and Systems Security Lab	IT4171N	0	0	3	2	3	50
		Practical Sub-total		0	0	3	6	3	250
		Seventh Semester Total		12	0	3	18	15	650

PSE for Seventh Semester	Course Code	OE for Seventh Semester	Course Code
1. Distributed Algorithms	IT4121N	1. Mobile Computing	IT4161N
2. Pattern Recognition	IT4122N		
3. Cloud Computing and Web Services	IT4123N		
4. Parallel and Distributed Systems	IT4124N		
5. Bioinformatics	IT4125N		
6. Internet Technology	IT4126N		
7. Deep Learning	IT4127N		
8. Data Sciences	IT4128N		
9. Intelligent Transportation and Smart Systems	IT4129N		
10. Cyber Physical Systems and Security	IT4130N		
11. CAD for VLSI	IT4131N		
12. Cognitive Radio Networks	IT4132N		

<i>Eighth Semester</i>									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	OE	OE5 (from NPTEL for the students opting Internship/ for others from Institute)	IT426xN	3	0	0	3	3	100
		Theory Sub-total		3	0	0	3	3	100
2	I	One Semester Internship	IT4291N				2		50
3	O	Grand Viva	IT4292N				2		50
4	P	Project - II	IT4293N				4		200
		Practical Sub-total		0	0	0	8	0	300
		Eighth Semester Total		3	0	0	11	3	400

Second Semester

Course Code	IT1201N	Course Name	Discrete Mathematics and Graph theory	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Preliminary Concepts of SET Theory</i>	Co-requisite Courses	<i>Engineering Mathematics - II</i>	Progressive Courses	<i>Formal Languages and Automata Theory</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To understand foundational concepts of computing
------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Logic and Proofs :Propositions, Conditional propositions and Logical Equivalence, Predicate calculus, quantifiers, Normalization of well-formed-formulas, Method of proofs, mathematical induction	8	Understanding of formal logic
2	Language of Mathematics : Sets, Countable and uncountable sets, Axiomatic set theory, Relations, Equivalence relations, Partially ordered relation, POSET, well order sets, quasi order sets, lattice. Application to relational Databases, Functions, Inverse and composition of functions, one-to-one correspondence.	12	Understanding of Set Theory
3	Algebraic structures : Algebraic structures with one binary operation - semigroups, monoids and groups. Free and cyclic monoids and groups, permutation groups, normal subgroups. Algebraic structures with two binary operations - rings, integral domains and fields. Boolean algebra and Boolean ring.	14	Understanding Basics of Discrete Structure
4	Counting methods : Basic principles of counting (Inclusion- exclusion, addition and multiplication rules), permutations and combinations, algorithms for generating permutations and combinations, binomial coefficients and combinatorial identities, The pigeonhole principle. Introduction to Polya's theory of counting.	6	Understanding combinatorics
5	Recurrence relations : Introduction, recursively defined sequences, solving recurrence relations: the characteristic polynomial and generating functions. Applications to analysis of algorithms.	6	Understanding recurrence

6	Graph theory : Introduction to graphs and their basic properties: degree, paths and cycles, subgraphs, isomorphism, Euler and Hamiltonian paths and cycles, representation of graphs, connected graphs, planar graphs. Basic graph searching algorithms: BFS and DFS. Basics of tree and spanning tree.	6	Understanding graphs
7	Coloring of Graph : Graph coloring basics, chromatic number, 4-color problem.	4	Understanding graphs
	Total	56	

Course Outcome	Understanding of foundational concepts of computing
Learning Resources	1. Discrete Mathematics and its Applications by Kenneth H Rosen, PHI 2. Discrete MATHEMATICS FOR Computer Scientists, J L Mott, A Kandel, and T P Baker 3. Concrete Mathematics: A Foundation for Computer Science, by Ronald Graham, Donald Knuth, and Oren Patashnik 4. Graph Theory With Applications To Engineering And Computer Science, Narsingh Deo, Tata Mc Graw Hill 5. Graph Theory, F Harary, Narosa

Course Code	IT1271N	Course Name	Introduction to Departmental Labs (Fundamentals of Programming)	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Introduction to Computing, Computer Programming Practice Lab</i>	Co-requisite Courses	NA	Progressive Courses	<i>Data Structures and Algorithms, Foundations of Programming Languages and Translation</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	NA

Course Objective	This laboratory course aims to strengthen students' ability to apply C programming constructs to complex problem-solving scenarios without the use of advanced data structures or graph theory. By engaging with computationally intensive and logically challenging assignments, students will develop: – A deeper understanding of fundamental C constructs (control structures, arrays, pointers, functions, macros, file handling, recursion, and bitwise operations). – Skills in algorithm design, numerical computation, code optimization, and debugging. – The ability to apply programming knowledge to solve domain-specific engineering problems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Module 1: Mathematical & Computational Programming – High-precision factorial calculation (up to 200 digits) using arrays. – Prime factorization with optimized trial division and bitwise operations. – Sieve of Eratosthenes using bit-level storage. – Integer-based floating-point multiplication simulation. – Gaussian elimination for solving linear equations. – Modular exponentiation using bitwise shifts. – Taylor series approximation of $\sin(x)$ and $\cos(x)$. – Numerical integration (Simpson's and trapezoidal rule). – Newton-Raphson method for square roots. – Recurrence relation computation with memoization.	15	– Apply control structures, loops, and arrays to solve large-number and high-precision computational problems. – Implement mathematical algorithms (factorization, numerical methods, series expansion) using efficient C programming techniques. – Optimize numerical computations using bitwise operations for speed and memory efficiency. – Evaluate approximation accuracy of numerical methods against analytical results.
2	Module 2: String and Character Processing – Custom <code>printf()</code> implementation. – Reverse words in a sentence without extra arrays. – Comment removal from a C source file. – Manual substring search. – Number-to-words conversion. – Caesar cipher encryption/decryption.	9	– Manipulate strings and characters in memory without relying on advanced libraries.

	– Palindrome checking (ignoring spaces, punctuation, and case).		– Implement low-level string parsing and transformation algorithms. – Develop secure text-processing routines for encryption, decryption, and pattern matching. – Apply character encoding logic to solve text-formatting and validation problems.
3	Module 3: Bitwise Programming – Counting set bits in an integer. – Swapping variables without a third variable or arithmetic ops. – Gray code generation. – Bit rotation (left and right). – Parity check using XOR.	6	– Demonstrate mastery of bitwise operators to implement compact and efficient algorithms. – Apply bit-level manipulations for encoding, rotation, parity checks, and logical transformations. – Optimize variable manipulation using bit tricks instead of arithmetic or extra memory. – Develop understanding of low-level data representation in C.
4	Module 4: Recursion & Backtracking – Generating permutations of a string. – Generating all binary strings of length n. – Tower of Hanoi solution and move listing. – N-Queens problem with array-based board representation.	6	– Apply recursion for generating permutations, combinations, and mathematical sequences. – Design recursive

			solutions for classic computational problems (Tower of Hanoi, binary string generation). – Implement backtracking logic to explore solution spaces efficiently (e.g., N-Queens). – Compare recursive vs. iterative approaches in terms of complexity and memory usage.
5	Module 5: File Handling Applications – Merging multiple text files. – File content analysis (line, word, char count). – Timestamped log system. – Search and replace in a file without a temporary file.	3	– Perform structured file I/O operations (reading, writing, appending) in C. – Implement programs for automated text manipulation, search, and replacement in files. – Handle file data efficiently without creating unnecessary intermediate storage. – Apply file-based processing to simulate basic system utilities (merge, log, wc).
6	Consolidation and Evaluation.	3	
	Total	42	

Course Outcome	Upon successful completion of this course, students will be able to: CO1: Apply advanced features of the C language to design efficient solutions for engineering-related computational problems. CO2: Develop modular programs using functions, recursion, and pointer-based techniques. CO3: Implement numerical, mathematical, and string-based algorithms with precision handling. CO4: Apply file handling operations for data processing and manipulation. CO5: Optimize code for performance using bitwise operations and low-level programming logic. CO6: Analyze, debug, and test programs systematically for correctness and efficiency.
-----------------------	--

Learning Resources	– Kernighan, B.W., Ritchie, D.M. – The C Programming Language, 2nd Ed., Prentice Hall. – Kanetkar, Y.P. – Let Us C, BPB Publications. – Deitel, P.J., Deitel, H.M. – C: How to Program, Pearson. – Herbert Schildt – C: The Complete Reference, McGraw-Hill.
---------------------------	---

Third Semester

Course Code	IT2101N	Course Name	Data Structure and Algorithms	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Concepts of C Language</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Analyze the time and space complexity of algorithms Understand the performance of various data structures Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees Understand different sorting and searching techniques
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Functions; arrays; introduction to pointers; structures; dynamic allocation; linked structures; time and space requirements. Algorithm Analysis, Asymptotic notations, Running Time Calculations	4	Familiar with programming and algorithm
2	Stack: Abstract Data Types (ADTs), Implementation of vector, Array Implementation Multiple Stacks, Applications and use of Stacks: Conversion from Infix to Postfix, Evaluation of Postfix Expressions, Prefix Notation, etc.	8	Understanding the linear data structure array, stack
3	Queue: Introduction, Liner Queue, Circular Queue, De-queue, Priority Queue, Array Queue Implementation, Applications of Queues, General List.	6	Understanding queue and its application
4	Linked Lists: Introduction, pointer and Implementation, Linear Linked Lists, Circular Linked Lists, Doubly Linked Lists, Doubly circular, Implementation of Linked Lists, Linked Stacks and Queues, Application of Linked List: Polynomials, High precision Arithmetic, Josephus Problem, etc.	8	Understanding link list and its application
5	Recursion: Recursion Algorithm, Type of Different Recursion Algorithms, Removal of Recursion.	2	Use of recursion

6	Trees & Graph: Tree Terminology, Binary Tree, Binary Tree Representation, Binary Tree Traversals, Threaded Binary Tree, Binary Search Tree Concepts and Implementation. Heap tree, AVL Tree, Red-Black tree. Introduction, BFS and DFS, connected components, spanning trees, shortest paths, max flow.	12	Understanding tree graph and its implementation and traversal
7	Hashing: Insert, search, delete, collision resolving techniques	4	Understanding hashing
8	Search Methods: Linear search, Binary search, Complexities of the searching algorithms	4	Analysis of search methods
9	Sorting: Introduction to sorting and comparison of Sorting Techniques, Complexities of sorting algorithm.	8	Implementation and analysis of sorting techniques
	Total	56	

Course Outcome	Choose the appropriate data structure and algorithm for a specified application Learn linear data structures such as arrays, linked lists, stacks, and queues and analyze their performance. Implement non-linear data structures like trees and graphs Implementation and analysis of sorting and searching techniques
-----------------------	---

Learning Resources	1. Seymour Lipschutz, Data Structures, Schaum's Outlines Series, Tata McGraw-Hill. 2. Ellis Horowitz, SatrajSahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, W. H. Freeman and Company. 3. Goodrich, Michael T. & Roberto Tamassia, Algorithm Design, Wiley Singapore. 4. Cormen, Thomash H., Leiserson, Charles E., Rivest, Ronald L., & Stein, Clifford. Introduction to Algorithms.
---------------------------	--

Course Code	IT2102N	Course Name	Digital Logic and Circuit Design	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Basic Electrical and Electronic Engineering</i>	Co-requisite Courses		Progressive Courses	<i>Computer Organization and Architecture, Hardware System Design, Digital System Design</i>
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	1. To learn Boolean algebra and number systems 2. To understand the steps for design of combinational and sequential circuits 3. To learn logic minimization and optimization techniques 4. To gain knowledge for analysis of timing and identify circuit hazards 5. Understand basics of memory, programmable logic, and Data Converters
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Number systems and Codes : Number representation and Computer arithmetic codes	2	understand and apply various number systems, codes, and computer arithmetic techniques
2	Boolean Algebra and Minimization Techniques :Boolean Logic operations, Basic laws, De Morgan's theorems, SOP/POS, K-map, Quine-McCluskey or Tabular method of minimization .	6	Learn Boolean Algebra and Boolean logic minimization technique.
3	Logic Gates :Logic Gates, Mixed Logic, Multilevel Gating networks, Multilevel output gate networks.	6	understand basic and mixed logic gates, including multilevel gating and multilevel output gate networks.
4	Logic Families :Digital Integrated Circuits, Introduction to logic families, CMOS logic .	4	Learn implementation of logic gates using CMOS logic.
5	Combinational Circuits : Multiplexers, Demultiplexers, Decoders, Encoders, Parity Generator/Checker, Code converters, Magnitude comparators, Applications	8	Understand the design of combinational logic blocks
6	Arithmetic Circuits : Adders and Subtractors, Binary multiplier, Binary divider	6	Understand the design of Arithmetic Circuits
7	Sequential Blocks: Latches, Flip-Flops (Clocked SR, JK, D, T), Triggered Flip-Flops,	6	Understand the design of Sequential logic blocks

	Realization of one flip-flop using other, Flip-flop ICs, Applications		
8	Counters :Asynchronous (Ripple or Serial) counters, Synchronous (Parallel) counters, Applications	4	Understand digital counters using basic sequential blocks
9	Registers : Universal shift registers, Shift register counters, Sequence generator	2	Understand digital registers using basic sequential blocks
10	Memory devices : Classification, Basic memory structure, ROM, RAM, Memory decoding, Memory expansion, PLD	4	Understand basics of memory, programmable logic
11	D/A and A/D converters :Analog and digital data conversions, Basic D/A conversion techniques (weighted resistor, R-2R ladder type etc.), Different A/D converters (Successive approximation, Single slope, Dual slope)	4	Learn basics of Data Converters and design of data converters
12	Applications of Digital Circuits :Frequency counter, Dot matrix display system, Digital multimeter etc.	2	Develop practical applications using Digital circuits
13	Testing issues in the Digital Circuits	2	Understand basic defects and testing techniques in Digital circuits
	Total	56	

Course Outcome	1. Understand number systems, binary arithmetic, and various digital codes used in digital systems. 2. Apply Boolean algebra and minimization techniques to simplify logic expressions and optimize digital circuits. 3. Design and analyze combinational logic circuits using standard logic gates and design tools. 4. Design and implement sequential circuits such as flip-flops, counters, and finite state machines. 5. Interpret and construct timing diagrams, and evaluate hazards and propagation delays in digital circuits. 6. Develop basic skills for digital hardware systems.
-----------------------	--

Learning Resources	1. Logic and Computer Design Fundamentals: by Mano, Kime: Pearson 2. Modern Digital Electronics: by Jain: TMH 3. Digital Design: by Mano 4. Digital Fundamentals: by Floyd, Jain: Pearson 5. Digital Circuits and Design: by Salivahanan, Arivazhagan: Vikas 6. Digital Principles and applications (5th Edition) : Leach & Malvino 7. Digital Computer Electronics : Malvino
---------------------------	---

Course Code	IT2103N	Course Name	Signals and Systems	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	<i>Functional analysis, Vector Algebra, Probability and Statistics</i>	Co-requisite Courses	<i>Digital Logic & Circuit Design, Math III</i>	Progressive Courses	<i>Communication System, Digital Signal Processing, Image Processing</i>
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Classification and representation, concepts of linear vector space and orthogonal signal representation, classification and properties of signals, System properties: linearity: additivity and homogeneity, shift-invariance, causality, stability, realizability	6	Familiarization with signals, systems, analogy with signals and vectors, vector-signal representation
2	Fourier Series, Fourier Transform and its properties	6	Frequency description of signals
3	Parseval's theorem, Bandwidth of signals, duality of time and frequency representations of signals.	2	Power and energy calculation, bandwidth calculation
4	Discrete time signal: sampling, digitization and reconstruction of analog signals, DTFT and DFT.	6	Discrete signal, frequency domain analysis, fast implementation
5	Introduction to random signals and their properties: random variables and processes for characterization and analysis of message signal and noise, statistical average, moments, mean square error, estimation of random variables	6	Random variable and statistical characteristics, random signal analysis
6	Random process, classification of random processes, geometric representation of random process, Gaussian random process, auto and cross-correlation, power spectral density, MMSE, LMSE	8	Random process, autocorrelation and cross-correlation studies

7	Introduction to system and classification, discrete time system, signal distortion in transmission, distortion-less conditions for signal transmission. Linear time invariant (LTI) system, impulse response, convolution, transfer function, Bandwidth of systems. System response to random signals.	8	LTI system, properties, random signal response of LTI system
	Total	42	

Course Outcome	To understand signals analysis and system response. Frequency domain description of signals, bandwidth estimation, energy and power spectrum density analysis. To understand the LTI system and capable of modeling and solving diverse problems in this framework. The course will help students to understand and solve problems of communications system, signal processing, image processing and multidisciplinary research problems.
-----------------------	---

Learning Resources	Text Books: 1. Linear Systems and Signals, B. P. Lathi, Oxford 2. Robert A. Gabel, Richard A. Roberts, "Signals and Linear Systems", John Wiley and Sons, 1995. 3. Probability and Random Processes with Applications to Signal Processing- H. Stark, J. W. Woods, Pearson Education Asia Reference Books: 1: A.V.Oppenheim, A.S.Willsky and S.H.Nawab -Signals & Systems, Pearson 2: S. Haykin & B.V.Veen, Signals and Systems- John Wiley
---------------------------	--

Course Code	IT2104N	Course Name	Formal Languages and Automata Theory	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Discrete Mathematics and Graph Theory	Co-requisite Courses	<i>Data Structures and Algorithms</i>	Progressive Courses		<i>Algorithms</i>
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards		

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Language and Grammar: definition, Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages	4	Understanding of formal grammar
2	Finite automata: Definition, Characteristics, Transitional system, deterministic finite automata (DFA), Nondeterministic finite automata (NFA), equivalence of DFA and NFA, Dead state, Finite Automata with output, Mealy machine and Moore machine, Conversion, Minimization of finite automata. Myhill-Nerode theorem, Two way finite automata, Application and limitation.	5	Understanding of finite automata
3	Finite State Machines: Definition, concept of sequential circuits, state table and state assignments, capability and limitations of FSM, state equivalence & minimization, Incompletely specified machine, Minimal machine, Merger graph, Merger table, Compatible graph.	4	Understanding of finite automata
4	Regular Expression: regular sets and regular expressions, Basic operations on regular expressions, Identities, Arden's Theorem, RE to NFA, ϵ -closure, NFA with ϵ move to DFA, Regular grammar from RE, pumping lemma for Regular expression, closure properties of regular expression, Decision problems of Regular expression, Application of RE.	5	Understanding of finite automata
5	Context-free languages and pushdown automata: Left and right linear grammars. Context-free grammars (CFG) and languages (CFL), parse trees, ambiguity in CFG, inherent ambiguity, Chomsky and Greibach normal forms, closure properties of CFL, pumping lemma for CFL, Application of context free grammar	4	Understanding of context free languages
6	Pushdown Automata(PDA), language recognized by PDA, deterministic and non deterministic PDA, equivalence of PDA and CFL, Multi stack PDA	4	Understanding of pushdown automata
7	Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages. Variations of Turing Machines, such as multi-tape Turing machine, non deterministic Turing machine, and their equivalence.	6	Stepping into Turing machines
8	Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.	3	Understanding of context sensitive languages
9	Computability in Turing machine, Church-Turing thesis, Computable functions, Undecidability, reducibility, halting problem, Time and space complexity of a Turing machine, Complexity classes, P and NP	7	Understanding computability
	Total	42	

Course Outcome	Understanding the mathematics of computing
-----------------------	--

Learning Resources	Books: Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Automata theory, languages, and computation. International Edition, 24, 19, TMH. Martin, J.C (2011). Introduction to Languages and The Theory of Computation, 4th Edition, TMH. Mishra, Chandrasekharan, Theory of Computer Science: Automata, Languages and Computation, 3rd Edition , PHI. Kozen, Dexter C. Automata and computability. Springer Science & Business Media, 2012. Sipser, Michael. "Introduction to the Theory of Computation." ACM Sigact News, 1996
---------------------------	--

Course Code	IT2171N	Course Name	Data Structure and Algorithms Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Concepts of C Language</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>		Data Book / Codes/Standards		

Course Objective	Provide practical implementation and usage of algorithms and data structures Identify the appropriate data structure to solve a given problem design and analyze simple linear and non linear data structures To implement and analyze the searching and sorting algorithms
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Pointer, array, structure and union	6	Understanding basics of programming
2	Stack and Queue	6	Implementation of stack and queue
3	Linked Lists	6	Implementation of linked list and its application
4	Recursion and Binary Tree	12	Implementation and analysis of binary tree
5	Search Methods	6	Implementation and analysis of searching methods
6	Sorting	6	Implementation and analysis of various sorting methods
	Total	42	

Course Outcome	Implement various data structures and their related operations Ability to solve problems using appropriate data structures Ability to implement sorting and searching algorithms using relevant data structures
-----------------------	---

Learning Resources	References: 1. Seymour Lipschutz, Data Structures, Schaum's Outlines Series, Tata McGraw-Hill. 2. Ellis Horowitz, Satraj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, W. H. Freeman and Company. 3. Goodrich, Michael T. & Roberto Tamassia, Algorithm Design, Wiley Singapore. 4. Cormen, Thomash H., Leiserson, Charles E., Rivest, Ronald L., & Stein, Clifford. Introduction to Algorithms.
---------------------------	--

Course Code	IT2172N	Course Name	Digital Logic and Circuit Design Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Digital Electronics Fundamental</i>	Co-requisite Courses		Progressive Courses	<i>Digital Hardware system Design</i>
Course Offering Department	<i>Information technology</i>			Data Book / Codes/Standards	

Course Objective	1. Convert theoretical concepts through practical implementation of digital circuits 2. Design and test basic combinational and sequential logic circuits using logic gates, digital ICs 3. Develop preliminary understanding of digital design using simulation tools and Hardware Description Languages (HDLs) like Verilog/VHDL 4. Analyze circuit behavior using timing diagrams and troubleshoot faults 5. Gain proficiency in implementing logic functions on breadboards or programmable devices (e.g., FPGA) 6. Encourage development of digital systems for problem-solving through hands-on
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction. Rules and precautions for hardware laboratory experiments. Demonstrations of basic logic building blocks, trainer kits, IC and components handling.	6	Learn lab safety rules and proper handling of trainer kits, logic ICs, and basic digital components.
2	Hands-on experiments and verification of the behavior of different logic building blocks.	9	Able to implement and verify basic logic circuits practically.
3	Logic simplification and minimization, design, implementation, and behavioral verification of different combinational logic units/functions.	9	Understand how to simplify, design, implement, and verify the behavior of various combinational logic circuits.
4	Design, implementation, and behavioral verification of different sequential logic units/functions.	9	able to design, implement, and verify the behavior of various sequential logic circuits.
5	Mini project (small group activity): Design, implementation, and verification of a large digital logic/arithmetic unit for real-world application.	9	Learn practical design, implementation, and verification of digital system for practical applications.
	Total	42	

Course Outcome	1. Learn lab safety and handle digital components correctly 2. Implement and verify basic logic gates and combinational circuits 3. Design and test sequential circuits like flip-flops and counters 4. Use HDLs and simulation tools for digital circuit modeling 5. Collaborate on projects to build and verify complex digital systems 6. Develop troubleshooting skills for digital circuit faults
-----------------------	---

Learning Resources	1. Tools for Logic circuit simulation 2. ModelSim or Vivado Simulator (HDL simulation) 3. Digital Logic Trainer Kits 4. FPGA Development Boards (e.g., Xilinx Spartan or Intel/Altera boards) 5. Breadboards, logic probes, and basic electronic components
---------------------------	---

Course Code	IT2173N	Course Name	Object Oriented Programming Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	Basic knowledge of C	Co-requisite Courses		Progressive Courses	
Course Offering Department				Data Book / Codes/Standards	

Course Objective	Understand and Apply Object-Oriented Programming (OOP) Principles, Programming on Class, Object creation, method handling. Knowledge about Encapsulation, Inheritance. Implement Exception Handling and File I/O Operations, Enhance Problem-Solving and Logical Thinking Skills
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Programs on Java Basics and Control Structures, Class and object	3	Develop and execute Java programs using basic syntax and control structures.
2	Programs on Classes, Constructors, and Methods. Static and Final method, Method overloading	6	to design and implement Java classes and polymorphism
3	Program on Inheritance. single, multilevel, and hierarchical inheritance., Method overriding, Super keyword, runtime polymorphism using dynamic method dispatch.	12	concepts of inheritance

4	Program on Interfaces and Packages, User defined package	3	Concept of package and interfaces
5	Programs on Exception Handling and File I/O Programs using try, catch, finally, throw, and throws. Read/write data using FileReader, BufferedReader, FileWriter, etc.	6	Concept of exception handling and file handling
6	Simple GUI applications development using AWT or Swing components.	6	
7	Mini Project	6	
	Total	42	

Course Outcome	Understand basic Java syntax and control structures to write and execute simple Java programs. Apply object-oriented principles such as classes, objects, encapsulation, and constructors in program design. Implement inheritance and polymorphism to create extensible and reusable Java code. Use interfaces and packages to organize and modularize Java applications effectively. Handle exceptions and perform file I/O operations for building robust and interactive Java applications.
-----------------------	--

Learning Resources	1. Effective Java 3rd Edition, <u>Joshua Bloch</u> , Addison-Wesley Professional 2. Core Java 11th Edition, <u>Cay Horstmann</u>, Pearson 3. Java: A Beginner's Guide, Ninth Edition , <u>Herbert Schildt</u>, <u>McGraw-Hill Osborne Media</u>
---------------------------	---

Fourth Semester

Course Code	IT2201N	Course Name	Foundations of Programming Languages and Translation	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Introduce fundamental principles and paradigms of programming languages, Understand the syntax, semantics, and pragmatics of programming language design, Examine key concepts such as data types, control flow, scoping, binding, and memory management, Familiarize students with virtual machines, abstract machines, and runtime environments.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Models of computation- imperative, functional and logical model, Evaluation and classification of programming language, Language Implementation- Compilation, Interpretation, Virtual machine	7	Basic of the course
2	Lexical Analysis, Token, Symbol table, data type - primitive, composite and recursive data type	5	A brief covering of Compiler
3	Functional programming language: Basic, functions and immutability, Imperative Versus Functional Programming, Lambda Calculus, Recursive Functions . Loops vs recursion, Tail recursion, Currying, Anonymous Functions, Higher order function, Sequential Execution, Iteration,	7	Develop skills in functional programming including immutability, recursion, higher-order functions, and lambda calculus.
4	Object Oriented Programming language: Dynamic Linking, Polymorphism, Inheritance, Threading, Garbage Collection, Interfaces, Adapters, Auto boxing and unboxing, Exception handling	6	Acquire understanding of object-oriented features

5	Syntax of a programming language- parse tree, abstract syntax tree, Ambiguity, Top down and bottom up parsing.	7	Learn syntax representation of programming languages, including parse trees, ambiguity resolution, and parsing strategies.
6	Type binding, Type checking, Type conversion-implicit, explicit	2	Understand type systems in programming languages, including type binding, type checking, and conversions.
7	Compiling standard Lex, YACC, Function call, If-then-else, Loop, Intermediate code, Code optimization	4	A detailed on compiler structure for conditional statements, loops etc.
8	Virtual machine specification: Basic, Process VM , System VM, Real machine vs Abstract machine vs Virtual machine, VM components, Instruction types, Bytecode format and encoding, Stack-based execution model, Fetch-decode-execute cycle, Switch-case based opcode execution	4	Understand the architecture and execution process of virtual machines, including their types, components, and bytecode execution.
	Total	42	

Course Outcome	To explain the concepts, syntax, semantics, and paradigms of programming languages along with the principles of language design. To Apply formal grammar representations and translation techniques to design and analyze language processors. To implement basic components of translators such as lexical analyzers and parsers using suitable algorithms or tools.
-----------------------	--

Learning Resources	1. John C. Mitchell, Michael R. Garey, Albert Meyer, "Foundations for Programming Languages" MIT Press 2. Kent D. Lee, Foundations of Programming Languages, Springer 3. Roosta Seyed H., Foundations of Programming Languages Design and Implementation, Cengage Learning India 4. Robert Harper, Practical Foundations for Programming Languages, Carnegie Mellon University
---------------------------	---

Course Code	IT2202N	Course Name	Communication Systems	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	<i>Signals and Systems, Digital logic and circuit design</i>	Co-requisite Courses	<i>Algorithms</i>	Progressive Courses	<i>Mobile Communications, Wireless Network, Computer Network</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	The Primary objective of the course is to demonstrate various analog and digital communication methods, understanding of their operation principles, utility in practical cases and analysis of real communication channels in presence of noise.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
Analog Communication	Introduction to electrical communication, elements of a communication system	2	Concept and importance of electrical Communications
Analog Transmission and reception	Introduction to modulation, Different types of analog modulations, AM, FM, PM with demodulation techniques	4	AM and FM transmitter and receiver operation, system and circuit, bandwidth analysis
Performance analysis of Analog Transmission	Different types of analog modulations, Comparative studies and performance analysis, modulation efficiency, SNR effect	4	Performance comparison of AM and FM in presence of noise

Multiplexing	Multiplexing techniques, Frequency division multiplexing (FDM) and quadrature carrier multiplexing (QCM), application of multiplexing	2	How to transmit and receive multiple message through spectrum access
Waveform coding-analog and digital conversion	PCM, DPCM, Delta and Delta sigma modulation, performance studies. Time division multiplexing and synchronization	4	Different analog to digital conversions for commun. and relative merits/demerits
Line coding	Baseband shaping for data transmission, properties and types, RZ, NRZ-ON-OFF, Polar, Bipolar, Manchester coding	4	Digital data transmission signalings and relative performance
ISI problem and mitigation	Nyquist criterion for ISI: zero and controlled ISI, eye pattern, equalization	4	Transmission impairments over finite bandwidth channel and Remedies
Digital Baseband Receiver	Integrate and dump receiver, probability of error analysis, Optimum filter, Matched filter and Correlator	4	Different receivers and relative performance analysis on binary signal reception
Digital carrier Modulation & Demodulation	Digital carrier modulation techniques: ASK, FSK, PSK, QPSK, MPSK, Coherent and non-coherent detection	6	Digital bandpass data transmission and reception
Noise	PSD and Bit vs symbol error probability and bandwidth Efficiency	2	Bandwidth analysis, probability of error analysis
Broadband Communication	Introduction to Spread Spectrum modulation, effect on thermal noise, single-tone interference and jamming, process gain	2	Data transmission through band spreading: interference mutation and avoidance techniques
Broadband Communication	Properties and generation of spreading code patterns, DSSS, FHSS, THSS techniques and their comparison	4	Different SS transmission and reception techniques
	Total	42	

Course Outcome	CO1: Students will know about various Analog modulation techniques and their comparative performance analysis CO2: Students will know about various Line Coding techniques and their comparative performance analysis CO3: Students will know about various Digital modulation techniques and their comparative performance analysis CO4: .Noise performance analysis on different baseband and bandpass digital communication schemes CO5:Students will also learn broadband communication and different spread spectrum techniques.
-----------------------	---

Learning Resources	• Modern Digital and Analog Communication Systems, B.P. Lathi , 5th edition, Oxford University Press, USA. • Communication Systems, A. B. Carlson, 4th Edition McGraw Hill 2002 • Communication Systems, Simon S. Haykin, John Wiley & Sons, 1983 • Principles of communication Systems, Taub Schilling, Tata Mc Graw Hill • Digital and Analog communication Systems, K. Sam Sanmugam • Digital Communications, J. G. Proakis, M. Salehi, 5th edition, McGraw-Hill, New York • Modern communication systems and spread spectrum, George R. Cooper, Clare D. McGillen, McGraw-Hill
---------------------------	---

Course Code	IT2203N	Course Name	Computer Organization and Architecture	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>IT 2102 (Digital Logic and Circuit Design) or any equivalent course</i>	Co-requisite Courses		Progressive Courses	<i>Advanced Computer Architecture /Embedded System Design/VLSI System Design</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	1. Understand fundamental concepts of computer organization and architecture 2. Learn data representation and instruction set formats 3. Analyze processor design including datapath and control units 4. Explore memory hierarchy and caching techniques 5. Study input/output systems and interfacing methods 6. Develop skills to evaluate and optimize computer system performance 7. Gain exposure into modern processor features like pipelining and parallelism
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Computer function and Interconnection: Computer Components, Computer function, Interconnection structures, Bus interconnection, PCI	3	Learn the basics of Computer function and Interconnection
2	CPU Arithmetic: ALU, Integer representation, Integer arithmetic, Floating point representation, Floating point arithmetic.	4	Familiar with the CPU arithmetic operations
3	Instruction Sets: Machine instruction characteristics, types of operands, Types of operations, Assembly language, Addressing, Instruction formats.	6	Learn the detailed instruction set architecture.
4	Processor Design and Datapath: Processor role, processor design goals, processor design process, datapath organization, main memory interface, local storage/register file, datapath for simple instructions, floating point unit datapath, advanced processors and datapaths.	8	Understand the details of processor and datapath design.
5	Processor design and control unit: Role of control unit, reset sequence, interrupt recognition and servicing, abnormal situation handling, instruction cycle and decisions involved, hardwired control unit, microprogrammed control unit.	7	Explain details of processor and control design.
6	Memory: Overview of computer memory system, memory parameters, classification of memory, main memory allocation, static RAM IC, Dynamic RAM, ROM logic, multiple memory decoding.	5	Familiar with the memory overview and its classifications.
7	Cache: Cache memory principles, elements of cache design, Cache organization.	3	Understand the fundamentals of cache memory and its implementations.
8	Internal memory: Semiconductor main memory, error correction, advanced DRAM organization.	3	learn the basics of internal memory and its implementations.
9	External memory: Magnetic disk, RAID, optical memory, magnetic tape.	3	Understand the external memory details and its classifications.
10	Input/Output: External devices, I/O modules, Programmed I/O, Interrupt driven I/O, DMA, I/O channels and processors, External interface: Firewire, Infiniband, and USB.	6	Learn the basics of input output devices and its interconnection mechanism with CPU.
11	Concurrency in Pipelining and Vector processing: Performance enhancement strategies, classification of parallelism, multiple functional units, pipelining, vector computing, array processor.	8	understand and apply pipelining, vector processing, and parallelism to enhance computer performance.
	Total	56	

Course Outcome	1. Learn basic structure and functioning of computer systems 2. Understanding of the representation of data and instructions in various formats and number systems 3. Get exposure on design and analyze processor datapaths and control units 4. Understanding of memory hierarchy, cache organization, and virtual memory concepts 5. Learn input/output mechanisms and interfacing techniques 6. Familiar with performance evaluation of computer architectures and identify bottlenecks 7. Get exposure on principles of pipelining, parallelism, and advanced processor features in system design
-----------------------	--

Learning Resources	Suggested Reading: 1. Computer Architecture and Organization Design Principles and Applications: B. Govindarajalu: TMH 2. Computer Organization and Architecture Designing for Performance: William Stallings: Pearson 3. Computer Architecture A Quantitative Approach: John L. Hennessy and David A. Patterson: ELSEVIER 4. Computer Systems Architecture A Networking Approach: Rob Williams: 2nd Ed: PEARSON 5. Computer Organization and Design The Hardware Software Interface ARM Edition: David A. Patterson and John L. Hennessy: MK
---------------------------	--

Course Code	IT2204N	Course Name	Design and Analysis of Algorithms	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Basic knowledge on Data Structures and Graph theory</i>	Co-requisite Courses	NA	Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards		

Course Objective	To enable the students with the ability to design efficient algorithms for computational problems, analyze their correctness and complexity, and apply appropriate algorithmic strategies such as divide-and-conquer, greedy methods, dynamic programming, and graph algorithms. It aims to develop skills in selecting suitable data structures, evaluating trade-offs between time and space, and applying mathematical tools to rigorously analyze performance.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Models of Computation; Algorithms and Complexity; Best case, worst case and average case; asymptotic notations	8	Understand fundamental models of computation. Distinguish between different algorithmic performance measures, asymptotic notations (O , Ω , Θ). Analyze algorithmic problems using formal complexity analysis techniques.
2	Sorting and searching; search trees; balanced trees; hashing	8	Implement and analyze standard sorting and searching algorithms. Compare the efficiency of various searching and sorting strategies.
3	Advanced data structure: Red-Black trees, Fibonacci heaps, data structure for disjoint sets	6	Understand the properties and operations of Red-Black trees and Fibonacci heaps. Analyze the amortized complexity of advanced data structures.
4	Lower bound theory	4	Understand lower bound concepts in

			algorithm analysis. Derive lower bounds for comparison-based sorting and other problems. Evaluate the theoretical limits of algorithm performance.
5	Optimization problems; Dynamic programming and Greedy method; theoretical foundations of greedy method	8	Formulate computational problems as optimization problems. Apply dynamic programming techniques to solve problems with optimal substructure.
6	Amortized Analysis, BFS and DFS, connected components, spanning trees, shortest paths, max flow	8	Implement and analyze BFS and DFS traversal algorithms. Apply algorithms for minimum spanning trees, shortest paths, and maximum flow problems.
7	Randomized algorithms: identity testing, primality and min cut	6	Understand the principles and benefits of randomized algorithms. Implement randomized algorithms for identity testing and primality testing.
8	Introduction to complexity classes: P, NP and NP completeness	8	Understand the definitions of complexity classes P and NP. Classify problems as belonging to P, NP, or

			NP-complete categories. Discuss the implications of the P vs. NP problem in computational theory.
	Total	56	

Course Outcome	1. Understanding fundamental models of computation, algorithmic performance measures, and asymptotic notations to evaluate algorithm efficiency. 2. Design and implement efficient algorithms using appropriate data structures such as balanced trees, hashing, and advanced structures like Red-Black trees, Fibonacci heaps, and disjoint sets. 3. Implement and evaluate graph algorithms for traversal, shortest paths, spanning trees, connected components, and network flow problems. 4. Classify problems into complexity classes (P, NP, NP-complete) and use polynomial-time reductions to prove NP-completeness.
-----------------------	---

Learning Resources	1. Aho, J. Hopcroft and J. Ullman; The Design and Analysis of Computer Algorithms, A. W. L, International Student Edition, Singapore. 2. S. Baase: Computer Algorithms: Introduction to Design and Analysis, 2nd ed., Addison Wesley. 3. T. H. Cormen, C. E. Leiserson and R. L. Rivest: Introduction to Algorithms, Prentice Hall of India, New Delhi. 4. E. Horowitz and S. Sabni: Fundamental of Computer Algorithms, Galgotia Pub. /Pitman, New Delhi/London 5. K. Mehlhom: Data Structures and Algorithms, Vol. 1 and Vol. 2, Springer-Verlag, Berlin, 1984. 6. A. Borodin and I. Munro: The Computational Complexity of Algebraic and Numeric Problems, American Elsevier, New York, 1975. 7. D. E. Knuth: The Art of Computer Programming, Vol. 1, Vol. 2 and Vol. 3. Vol. 1, 2nd ed., Narosa/Addison-Wesley, New Delhi/London, 1973; Vol. 2: 2nd ed., Addison-Wesley, 18 London, 1981; Vol. 3: Addison-Wesley, London, 1973. 8. S. Winograd: The Arithmetic Complexity of Computation, SIAM.
---------------------------	---

Course Code	IT2261N	Course Name	Multimedia Systems	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology			Data Book / Codes/Standards	

Course Objective	Foundational understanding of multimedia systems and technologies, Knowledge of various multimedia data types and their representations, techniques for compression, transmission, and retrieval of multimedia data, Explore multimedia system design including synchronization, storage, and delivery.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction to Multimedia , Elements of Multimedia, Properties of multimedia system, Categories, Features, Application, Phases of multimedia application development, Multimedia development team, Convergence of Multimedia System.	2	Basic of Multimedia
2	Text ASCII, Unicode (UTF-8, UTF-16), Raster vs. vector fonts, Text rendering: Anti-aliasing, kerning, ligatures, Natural Language Processing (intro): Tokenization, stemming, lemmatization Text Mining and Information Extraction	3	Knowledge of text and natural language processing
2.	Image: Raster and Vector, Types of image, Digital image representation, Binary, Grayscale and color image, Image negation, change of dynamic range, Intensity level slicing, Histogram, Color model RGB, CMYK, HSB, HSV, CIE-lab, Conversion from one color model to other, File system (TIFF, BMP, PCX, GIF etc.), Display devices, CRT Monitor, LCD monitor, PDP, Touch screen- Resistive and capacitive, Surface acoustic wave, infrared touch screen, Flexible display-epaper and OLED. Digital Scanner, Digital Camera, Printer-dot matrix, inkjet, laser, 3D printer. Different research areas in image processing-OCR, Image encryption, Steganography	8	Detailed concept of image, display device
3.	Compression: Advantages, disadvantages, Spatial and temporal redundancies, Different Lossless and Lossy compression techniques, Performance measurement, DPCM, Lampel-Ziv, Huffman coding, Adaptive Huffman coding, Arithmetic coding, GIF, JPEG.	6	Basic compression techniques
4.	Audio: Sound wave, Physical characteristic, Musical note, Components of Audio System, Microphone:moving coil, condenser, Amplifier :class A Class B, Speaker, Synthesizer, MIDI. Sound card, Digital	6	Knowledge on audio, audio

	Audio processing. Audio compression: PCM, ADPCM, MPEG audio standards, Audio watermarking and steganography, Music Information Retrieval, Speaker Recognition and Verification		compressi on
5.	Video: Luminance & Chrominance, Luma and Chroma, Chroma Sub- sampling, Television Systems PAL, NTSC, Video Nomenclature HDTV, EDTV, Video Quality and Performance Measurements, Streaming video, DTH, IPTV, Digital Video Processing, Video capture, Video processing AVO/AVI file formats, Video compression-I, B, P frame, MPEG, Multimedia synchronization- skew and jitter.	6	Knowledg e on video, video compressi on, file format
6.	Animation: Key frame and Tweening, Cell Animation, Rotoscoping, Stop-Motion Animation, Motion Cycling, Computer Based Animation, Path based animation, Client pull and server push, Virtual reality	3	Basic knowledge about animation
7	Multimedia Database-Image Representation, Segmentation, Similarity based retrieval, Image retrieval by color, Shape & texture, indexing – K-d-tree, R-tree, Video Content, Quad tree, Quarrying, Video Segmentation, Indexing.	4	Multimedi a database
8	Multimedia Network Fundamentals, Multimedia Protocols for the Internet, Multimedia Networking Services	4	
	Total	42	

Course Outcome	Understand the fundamentals of multimedia systems and components, Analyze audio, image, and video data representation and processing techniques, Apply appropriate compression techniques for multimedia data., Evaluate multimedia delivery mechanisms including streaming and synchronization, Develop simple multimedia applications using tools and APIs. Basic concept about multimedia database
-----------------------	---

Learning Resources	References: 1) R. Steinmetz, K. Nahrstedt, "Multimedia Systems", Springer Science & Business Media. 2) J.F.K, Buford, Multimedia Systems, ACM Press. 3) Sloane, Multimedia Communication, McGraw Hill. 4) Boyle, Design for Multimedia Learning Prentice Hall. 5) B Prabhakaran, Kluwer, Multimedia Database Management Systems, Springer.
---------------------------	--

Course Code	IT2271N	Course Name	Foundation of Programming	Course Category	PC	L	T	P
--------------------	---------	--------------------	---------------------------	------------------------	----	---	---	---

			Languages and Translation Lab			0	0	3
--	--	--	-------------------------------	--	--	---	---	---

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	1. Gain hands-on experience with lexical analysis, parsing, and translation. 2. Implement features of functional and object-oriented paradigms. 3. Use tools like LEX and YACC to design language processors. 4. Simulate compilation, interpretation, and virtual machine concepts.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Implement simple programs for functional, imperative and logical paradigms and recursion. Compare evaluation strategies: implement the same algorithm imperatively and functionally,	6	Basic
2.	Simulation of compilation vs interpretation using a small expression language.	3	Concept of compilation vs interpretation
3	Implement a basic virtual machine that executes arithmetic bytecode (stack-based).	6	Basic of Virtual Machine
4.	Implement a lexical analyzer in C/C++ for identifiers, keywords, and numbers.	3	Concept of lexical analyzer
5.	Implement lambda calculus expressions in a functional language.	3	Concept of lambda calculus
6.	Write examples of recursion, tail recursion, currying, higher-order functions, and compare with loops.	6	Recursion and tail recursion
7.	Implementation of LEX programming	6	Concept of LEX programming
8.	Simulate a stack-based VM with fetch-decode-execute cycle and switch-case opcode execution.	6	Stack based Virtual Machine
9.	Mini Project	3	
	Total	42	

Course Outcome	Apply functional, imperative paradigms in practical problems. Basic concept of Lexical Analyzer Use LEX to build basic translators.. Analyze and compare language implementation models.
-----------------------	--

Learning Resources	1. John C. Mitchell, Michael R. Garey, Albert Meyer, "Foundations for Programming Languages" MIT Press 2. Kent D. Lee, Foundations of Programming Languages, Springer 3. Roosta Seyed H., Foundations of Programming Languages Design and Implementation, Cengage Learning India 4. Robert Harper, Practical Foundations for Programming Languages, Carnegie Mellon University
---------------------------	---

Course Code	IT2272N	Course Name	Communication Systems Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Basic Electronics Lab, Digital Logic and Circuit Lab</i>	Co-requisite Courses		Progressive Courses	<i>Image Processing Lab</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Software (Matlab) simulation and Hardware realization of different signals, like impulse, step, triangle, ramp, square; Software and Hardware realization of AM and FM Software and Hardware realization of Line codes:On- OFF, Polar and Bipolar, RZ and NRZ Study of FDM and TDM signals (transmission and reception) Software and Hardware circuit of experiment on BASK, BFSK and BPSK
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Design and study of some basic communication signals	6	Software and Hardware generation of different communication signals
2	Study of sampling and reconstruction	6	Verification of Sampling Theory

3	Circuit design and study of various analog modulation techniques	6	Study on different AM and FM modulation and demodulation
4	Study of waveform coding techniques	6	Software and hardware generation of RZ, NRZ, ON-OFF, Polar and Bipolar signals
5	Experiments on digital modulation techniques ASK, FSK, PSK. Circuit design and study of behaviour of the circuits	9	Study transmission and reception of various digital communication-wired and wireless connection
6	Software and Hardware Generation of PRBS and verification of properties	6	Software simulation and generation on Hardware circuit of binary pseudo random sequence
7	Generation and Reception of PN Spread Spectrum	3	Spread spectrum signal transmission and reception
	Total	42	

Course Outcome	• To verify different concepts studies on Communication System Course. • Software and hardware realization, verification and performance studies on different analog and digital signal transmission • Multiplexing Techniques for analog and digital signal transmission
-----------------------	---

Learning Resources	• Laboratory Sheets • Lab Manuals • Modern Digital and Analog Communication Systems, B.P. Lathi , 5th edition, Oxford University Press, USA. • Taub's Principles of communication Systems, Tata Mc Graw Hill
---------------------------	---

Course Code	IT2273N	Course Name	Computer Organization and Architecture Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Digital logic design/Computer Organization and Architecture (Theory)</i>	Co-requisite Courses		Progressive Courses	<i>Advanced Computer Architecture /Embedded System Design/Digital VLSI Design</i>
Course Offering Department				Data Book / Codes/Standards	

Course Objective	1. Provide hands-on experience with computer hardware components and architecture concepts 2. Develop skills in designing and simulating processor datapaths and control 3. Implement and analyze memory systems, cache, and I/O interfacing 4. Develop basic skills in assembly language programming and processor operations 5. Design and simulate combinational and sequential circuits using Verilog/VHDL 6. Gain hands-on experience with Xilinx ISE and HDL-based hardware design 7. Implement and test computing units and processor elements on FPGA boards
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Know your Computer and its Organization, Hands on demonstration of assembling and disassembling of PC.	3	Able to identify computer components, understand their organization, and demonstrate skills in assembling and disassembling a PC.
2	Hands on experience with different components of computers of different generations, Basic troubleshooting with everyday usage of computers.	3	Hands on for identification of computer components and demonstrate PC assembly and disassembly skills.
3	Introduction to Verilog/VHDL and Xilinx ISE, Tutorial on Verilog/VHDL as Hardware Description Language, Tutorial with hands on demonstration in Xilinx ISE Design Tool (Programming Language: VHDL), Build your own Computing Units.	9	understand Verilog/VHDL basics, use Xilinx ISE for hardware design, and implement computing units through hands-on practice.
4	Experiments on different combinational design blocks and simulation using Verilog/VHDL under ISE environment.	6	design, simulate, and verify various combinational logic blocks using

			Verilog/VHDL in the Xilinx ISE environment.
5	Experiments on different sequential design blocks and simulation using Verilog/VHDL under ISE environment.	6	design, simulate, and verify various sequential logic blocks using Verilog/VHDL in the Xilinx ISE environment.
6	Experiments on designing different computing units for processing, memory, and IO interfacing.	6	design and implement computing units for processing, memory, and I/O interfacing using HDL and simulation tools.
7	Mini project (Group activity): A small project related to Computing Unit design and Simulation.	6	design, implement, and simulate a small computing unit project using Verilog/VHDL and appropriate simulation tools.
8	Verify and Test your design: Verification and test of simulated and synthesized design using FPGA Prototype Boards	6	verify and test their simulated and synthesized designs on FPGA prototype boards to ensure functional correctness and real-time performance.
	Total	42	

Course Outcome	1. Identify and assemble key hardware components of a computer system 2. Write and execute basic assembly programs for processor-level understanding 3. Design and simulate combinational and sequential circuits using Verilog/VHDL in Xilinx ISE 4. Understand and apply basic VHDL syntax and HDL-based hardware design concepts 5. Build and test simple computing units through simulation and implementation 6. Verify and debug digital designs on FPGA prototype boards
-----------------------	--

Learning Resources	1. Xilinx ISE/Vivado – For Verilog/VHDL simulation and FPGA synthesis 2. ModelSim – HDL simulation (VHDL/Verilog) 3. FPGA Development Boards: Xilinx Spartan-6, Artix-7, or Intel Cyclone IV 4. Logic ICs for manual logic circuit implementation 5. Breadboards, jumpers, logic probes, and test equipment (oscilloscope, multimeter)
---------------------------	--

Course Code	IT2274N	Course Name	Algorithms Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	The objective of this laboratory course is to enable students to: 1. Understand and implement fundamental and advanced algorithmic techniques for problem-solving. 2. Develop skills in designing, coding, testing, and analyzing algorithms for efficiency in terms of time and space complexity. 3. Apply appropriate data structures and algorithmic paradigms such as divide-and-conquer, greedy methods, dynamic programming, and backtracking to solve computational problems. 4. Foster problem-solving ability through the application of algorithms to real-world scenarios and competitive programming challenges. 5. Enhance programming proficiency and debugging skills using a high-level language (e.g., C/C++/Python) in an algorithmic context. 6. Encourage creativity in approaching problems by comparing alternative algorithmic solutions and optimizing performance.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Study of time requirements of searching and sorting algorithms; Tally the experimental time requirement with the theoretical time complexity; Understanding of problem size	8	Implement and execute standard searching and sorting algorithms. Measure execution time for various input sizes and compare with theoretical complexity.
2	Greedy, Dynamic Programming, Text file compression using Huffman coding Branch and Bound, Backtracking	8	Implement greedy and dynamic programming solutions for selected problems. Compare performance differences

			between greedy and dynamic programming approaches. Select suitable problem-solving paradigms based on problem characteristics.
3	Implementation of graph algorithms; Study of data structures' roles in developing efficient algorithms (in connection with graph algorithms)	10	Implement graph traversal, shortest path, spanning tree, and flow algorithms. Select appropriate graph representations (adjacency list/matrix) for efficiency. Demonstrate the impact of underlying data structures on algorithm performance.
4	Role of randomness in computing	8	Implement algorithms that use randomization for improved efficiency or simplicity. Evaluate the expected performance of randomized algorithms through experiments.
5	Implementation of some of the number theoretic algorithms	8	Apply number-theoretic algorithms to cryptographic or combinatorial problems. Analyze their performance for large inputs.
	Total	42	

Course Outcome	1. Implement, execute, and experimentally evaluate the performance of searching, sorting, and other algorithms, and compare results with theoretical time complexity. 2. Apply algorithmic paradigms such as greedy methods, dynamic programming, backtracking, and branch-and-bound to solve computational problems. 3. Develop applications involving Huffman coding for text file compression and evaluate compression efficiency. 4. Implement and analyze graph algorithms, demonstrating the role of appropriate data structures in improving performance.
-----------------------	---

Learning Resources	GCC / G++ / Clang – C/C++ compilers for implementing algorithms. Python (with libraries like networkx for graph algorithms, heapq for priority queues). Java (built-in collections framework for data structures). GNU time or built-in timing libraries – for empirical time measurement. GitHub / GitLab – for version control of algorithm code. Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. – <i>Introduction to Algorithms</i>, 4th Ed., MIT Press, 2022. (Covers analysis, paradigms, data structures, graph algorithms, and complexity) Kleinberg, J., and Tardos, É. – <i>Algorithm Design</i>, Pearson, 2005. (Greedy, dynamic programming, graph algorithms) Dasgupta, S., Papadimitriou, C.H., and Vazirani, U.V. – <i>Algorithms</i>, McGraw-Hill, 2008. (Simplified explanation with problem-solving focus) Sahni, S. – <i>Data Structures, Algorithms, and Applications in C++</i>, Universities Press, 2005. (Implementation-oriented approach)
---------------------------	---

Fifth Semester

Course Code	IT3101N	Course Name	Microprocessor and Microcontroller	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	<i>Digital Logic and Circuit Design (IT2102N) or equivalent course</i>	Co-requisite Courses	<i>Computer Architecture or Organization</i>	Progressive Courses	<i>Embedded System Design, Internet-of-Things, Real-Time Systems</i>
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	• To understand the architecture and working principles of microprocessors and microcontrollers. • To develop programming skills in assembly language for microprocessor and microcontroller-based systems. • To familiarize students with interfacing techniques of memory and I/O devices. • To introduce the concept of embedded systems using microcontrollers. • To enable students to design and implement real-time applications using microcontrollers.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to 8085A CPU: Pin description and features, architecture register organization.	2	Will learn the function of each pin of 8085, internal architecture and key functional blocks
2	8085 Addressing: Different addressing modes and their features, Instruction set, Instruction cycle, machine cycle, Timing diagram.	4	Will learn the addressing modes, concept of machine cycles and

			T-states, and execution timing diagram.
3	8085 Assembly Language Programming	4	Will learn the assembly level programming for 8085
4	Hardware Interfacing: Interfacing memory, peripheral chips (IO mapped IO and Memory mapped IO), Interrupts and DMA.	4	Will learn how interfacing is done with a microprocessor
5	16 bit processors: 8086 and architecture, segmented memory cycles, read/write cycle in min/max mode, Reset operation, wait state, Halt state, Hold state, Lock operation, Interrupt processing.	4	Will learn the architecture of 8086 and difference with 8085
6	8086 Addressing modes and their features: Software instruction set (including specific instructions like string instructions, repeat, segment override, lock prefixes and their use) and Assembly Language programming with the same.	6	Will learn the addressing modes, instruction sets and assembly level programming for 8086
7	8051 Microcontroller: Architecture, Memory Management, Instruction set and Assembly Language Programming.	6	Will learn the concept of 8051 microcontroller and its architecture and assembly programming
8	ARM RISC Architecture, Memory Management, Instruction set and Assembly Language Programming.	6	Will learn ARM RISC microcontroller

			and program ming
9	Overview of Raspberry Pi, and Arduino Controller and its application development.	6	Will learn to design and implement real-time applications using microcontrollers.
	Total	42	

Course Outcome	At the end of the course, students will be able to: • CO1: Describe the architecture, features, and operation of 8085/8086 microprocessors and 8051 microcontroller. • CO2: Develop and debug assembly language programs for microprocessor and microcontroller systems. • CO3: Interface memory and peripheral devices with microprocessors and microcontrollers. • CO4: Analyze timing diagrams, interrupt handling, and addressing modes in processors. • CO5: Design embedded system applications using 8051 and integrate them with sensors and actuators. • CO6: Compare and choose between different microcontrollers and microprocessors for a specific application.
-----------------------	--

Learning Resources	1. Microprocessors Architecture, Programming, and Applications with the 8085-R. Gaonkar 2. The Intel Microprocessors- Brey 3. Microprocessors and interfacing: Programming and hardware- Douglas V. Hall 4. The 8051 Microcontroller and Embedded Systems Using Assembly And C, 2/E-Mazidi (Pearson Education India). 5. Advanced Microprocessors and Peripherals - Ajoy Kumar Ray, K M Bhurchandi (TMH) 6. ARM System On Chip Architecture – Steve Furber
---------------------------	---

Course Code	IT3102N	Course Name	Database Management Systems	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Programming and Data Structure, Discrete Mathematics and Graph Theory</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department		<i>Information Technology</i>	Data Book / Codes/Standards		

Course Objective	To learn techniques and principles of managing database in large information systems
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction Database, Database Management Systems, Database Systems versus File Systems, View of Data, Database Languages, Database Users	4	Getting introduced to DBMS concepts
2	Components of a Database Management System, Data Independence, Network, Relational, Hierarchical, Object Oriented Data Models	4	Data Models
3	The Entity Relationship Model Basic Concepts, Constraints, Keys, Design Issues, Entity-Relationship Diagrams, Extended E-R Features, Relational Model - Structures of Relational Databases, Integrity Constraints, ER to Relational model	6	ER Modeling Concepts
4	Relational Query Languages Relational Algebra, Relational Calculus, and SQL.	8	Query Languages
5	Relational Database Design Functional Dependency, Armstrong's Axioms, Normal Forms, Dependency Preservation, Lossless design.	8	Design of RDMS
6	Storage Strategies Ordered, Unordered File, Hashing, Indexing, Single-Level, Multi-level Indexes, B tree and B+ tree	6	Database Storage
7	Query Processing Evaluation of Relational Algebra Expressions, Implementation of SELECT, JOIN, PROJECT Operations, Query Optimization Algorithms.	8	Query Processing & Evaluation
8	Transaction Processing Transaction concept, Schedule, Conflict & View Serializability, Concurrency Control, Lock base and Timestamp based Protocols, Multiversion and Optimistic Concurrency Control schemes.	8	Transaction Processing
9	Recovery Causes of failures, Immediate and Deferred Update, Shadow paging, Graph model (Neo4j), Advanced Database, Columnar Database, Key value, Document oriented database (No SQL [Mongo, Cassandra, Gremlin, Redis]), Time series database, Multi model	4	Failure Recovery
	Total	56	

Course Outcome	- Database system concepts - Data Models - Database Query Building - Database Storage Management
-----------------------	---

Learning Resources	1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Mc Graw Hill, 6th ed, 2013. 2. Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson, 7th ed, 2016. 3. C. J Date, "An Introduction to Database System", Pearson, 8th ed, 2003. 4. Ivan Bayross, "SQL, PL/SQL: The Programming Language of Oracle" BPB Publications, 4th ed, 2010
---------------------------	--

Course Code	IT3103N	Course Name	Operating Systems	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Computer Programming, Data Structure and Algorithms, Computer Organization and Architecture</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology		Data Book / Codes/Standards		

Course Objective	Students will learn 1. Basics of Operating Systems 2. Concepts of processes and threads 3. Concepts of CPU scheduling 4. Concepts of process synchronization 5. Concepts of deadlocks 6. Concepts of memory, device and file management
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Operating Systems, Overview, Evolution of Operating Systems, Basic architectural concepts, concepts of batch-processing, multiprocessing, multiprogramming, timesharing, real-time operations; interrupt handler	4	Students will learn basics of Operating Systems

2	Concept of process, Process synchronization, Process Management and Scheduling, Co-operating processes, Inter-process communication (IPC), Remote procedure call (RPC), Hardware requirements: protection, privileged mode, Threads and their management, Communication in client-server systems	6	Students will learn concepts of processes
3	Threads: Multithreaded model, scheduler activations, examples of threaded programs	3	Students will learn concepts of threads
4	Scheduling - CPU scheduling— short term, medium term and long term scheduling, non-preemptive and preemptive algorithms	7	Students will learn CPU scheduling algorithms
5	Process Synchronization - The critical section problem, Synchronization hardware, Semaphores, Classical problems of synchronization, Tools and constructs for concurrency	8	Students will learn process-synchronization techniques
6	Deadlocks - Modeling, characterization, Detection prevention and avoidance of deadlocks, Recovery from deadlocks, Dynamic resource allocation	6	Students will learn algorithms for detecting, preventing and avoiding deadlocks
7	Memory Management - Partitioning, paging, concepts of virtual memory, demand-paging, page replacement algorithms, working set theory, load control, segmentation, segmentation and demand-paging, Case studies, Windows. Current Hardware support for paging: e.g., Pentium/ MIPS processor etc.	10	Students will learn memory management techniques
8	Device Management - Scheduling algorithms -FCFS, shortest-seeK-time-first, SCAN, C-SCAN, LOOK, C-LOOK algorithms, Device drivers, concept of driver routines	4	Students will learn device management techniques
9	File Management - File concept, file support, directory structures, symbolic file directory, basic file directory, logical file system, physical file system, access methods, file protection, file allocation strategies	6	Students will learn file management techniques
10	Case Studies - UNIX/Linux, Windows, and Android	2	Case studies
	Total	56	

Course Outcome	On successful completion of this course, students will be familiar with 1. Basics of Operating Systems 2. Concepts of processes and threads 3. CPU scheduling algorithms 4. Process-synchronization techniques 5. Algorithms for detecting, preventing and avoiding deadlocks 6. Memory, device and file management techniques
-----------------------	--

Learning Resources	1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, John Wiley 2. William Stallings, Operating Systems: Internals and Design Principles. Prentice-Hall 3. AS Tanenbaum, Modern Operating Systems, 3rd Ed., Pearson 4. AS Tanenbaum, AS Woodhull, Operating Systems Design and Implementation, Prentice Hall 5. M. J. Bach. Design of the Unix Operating System, Prentice Hall of India 6. Harvey M. Deitel (Author), Paul Deitel (Author), David R. Choffnes (Author), Operating Systems, Pearson
---------------------------	---

Course Code	IT3121N	Course Name	Software Engineering	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To learn present-day Software Engineering principles, tools and techniques
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction Software Engineering Discipline and its Evolution, Software Projects vs. Products, Exploratory Style of Software Development, Human Cognitive Limitations, Emergence of Software Engineering.	3	Introduction
2	Software Life Cycle Models Documenting Software Life Cycle, Life Cycle Models, Classical Waterfall Model, Iterative Waterfall Model, Prototyping Model, Evolutionary Model, Spiral Model. Selecting suitable model for software development.	6	SDLC Models

3	Software Requirements Specification Requirements gathering, Requirements analysis, Software Requirements Specification (SRS), IEEE SRS format, Identifying functional and non-functional requirements, Documenting software requirements specification.	4	SRS Preparation
4	Function Oriented Software Design Design Activities and methodologies, Cohesion and Coupling in software design, Modular Software Design. SA/SD Methodology, Structured Analysis and Building DFD Models, Structured Design, Structure Chart Preparation, Transform and 6 Transaction Analysis.	6	DFD Modelling
5	Object Oriented Software Design Basic Object Oriented Concepts, UML Modelling, UML 2.x.	2	OOD
6	Coding Standards and Guidelines, Code Review Techniques: Code Walkthrough, Code Inspection, Software Documentation, Gunning's Fog Index.	3	Coding
7	Testing Basic software testing concepts, Test Case Generation, Black Box Testing, Equivalence Class Partitioning, Boundary Value Analysis, White Box Testing, Statement, Branch, Condition, Path coverage based testing strategies, Data flow based testing, Mutation Testing, McCabe's Cyclomatic Complexity, Integration Testing, System testing, Error Seeding, Regression Testing. Software testing metrics, Benefits, Metrics life cycle, Types , Process Metrics Product Metrics Project Metrics, Derivative Metrics, Defect Density, Defect Leakage, Defect Removal Efficiency, Defect Severity Index, Review Efficiency, Test Case Effectiveness, Test Case Productivity, Test Design Coverage, Test Execution Coverage, Test Tracking & Efficiency, Test Effort Percentage, Test Economic Metrics, Test Effectiveness, Test Team Metrics, Agile process metrics, Software Testing Key Performance Indicators (KPIs).	6	Software Testing
8	Reliability and Quality Management Software vs. Hardware Reliability, Reliability Metrics and Growth Models, Software Quality Management, Quality Systems, ISO-900, 9001, SEI CMM Model.	4	Software Reliability
9	Software Maintenance [3L] Software maintenance types, Software maintenance Characteristics, Reverse engineering, Maintenance process models.	4	Software Maintenance
10	Software Project Management [4L] Project planning, SPMP document, Project Size Estimation, Metrics, Estimation Techniques, Empirical, Heuristic, Analytical techniques, Scheduling, Project Monitoring, Activity networks and Critical Path Method (CPM).	4	Project Management
	Total	42	

Course Outcome	Learn complete Software Development Life Cycle and associated principles, tools and techniques
-----------------------	--

Learning Resources	1. R. Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2nd Ed, 2006. 2. R. S. Pressman, Software Engineering A Practitioner's Approach, Tata McGraw Hill, 6th Ed, 2005. 3. I. Sommerville, Software Engineering, Pearson, 7th Ed, 2005. 4. P. Jalote, An Integrated Approach to Software Engineering, Narosa, 2nd Ed, 1999.
---------------------------	---

Course Code	IT3122N	Course Name	Mobile Communications	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Communication systems, computer networks</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Students will know about wireless communication techniques, different generations of wireless communication (evolution to revolution) and some modern mobile communication techniques (5G and beyond)
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction: Introduction to mobile and radio communication, radio communication principles, ideas on transmitters, receivers, medium of signal propagation	2	Introduction to mobile communication, evolution to revolution
2.	Cellular concept: Frequently assignment, frequency reuse, concept of cell splitting, System capacity and interference	4	Spectrum assignment, sharing and access in cellular communication
3.	Mobile radio propagation: Multipath signal propagation model and signal fading in mobile environment, large scale path loss	4	Radio mobile channel characterization and propagation
4.	Mobile radio propagation: Small scale fading and multipath, Doppler effect	2	Multipath effect in radio mobile channel, concept of fading
5.	Receiver techniques for fading dispersive channels: Channel equalization, adaptive equalizing, diversity techniques.	2	Role of equalizer in mitigating fading effect
6.	Satellite link design: uplink, downlink, G/T ratio, (C/N), performance	4	Receiver performance satellite link analysis

7.	Multiple access schemes in mobile communication: TDMA, FDMA, CDMA, OFDM, Spread Spectrum Transmission and Reception	4	spectrum access techniques through orthogonal frequency, time code, space
8.	GSM architecture: Mobility management, Handover in cellular systems , Soft handover, hard handover, Security, international roaming for GSM, Mobile Number portability, SMS in GSM, VoIP service for Mobile Networks, GPRS architecture	6	GSM architecture , handover, macro-micro cell
9.	WLAN: architecture, ideas on mobile ad-hoc networking, protocols.	6	introduction to WLAN, functionality
10.	5G and beyond: mm wave communication, MIMO, small cell, beamforming, NOMA, MEC, WRAN, Reconfigurable Intelligent Surfaces	8	Promises and potentials of 5G and beyond, role and contributions of different technologies to make 5G attractive
	Total	42	

Course Outcome	CO1: Familiarization of techniques making evolution to revolution in mobile communication, familiarization to modern mobile communication techniques, handling the challenges on spectrum, security, energy (sustainability), service oriented design, green communications CO2: Modeling of Radio Mobile Channel and different fading CO3: To study different multiple access techniques CO4: To study GSM architecture and WLAN architecture CO5: 5G and B5G-techniques, promises and potentials
-----------------------	---

Learning Resources	Books: 1.T. S. Rappaport, “Wireless Communication: Principles and Practice”, PHI, Second Edition 2.Jochen Schiller, “Mobile Communications”, PHI, Second Edition, 2003. 3.Jochen Burkhardt, Pervasive Computing: Technology and Architecture of Mobile Internet Applications, Addison Wesley Professional; 3rd edition, 2007 4.T. Pratt, C. Bostian, “Satellite Communication”, 2nd Edition, John Wiley Co. 5.William Stallings, “Wireless Communications and Networks”, Pearson Education, 2002
---------------------------	---

Course Code	IT3123N	Course Name	Information and Coding Theory	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Communication Systems, Discrete math and graph theory</i>	Co-requisite Courses	<i>Communication Systems, Mobile Communication</i>	Progressive Courses	<i>Computer Networks, Internet of Things, Embedded Systems and IoT, Wireless Networks</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	To understand information theoretic behavior of a communication system. To understand various source coding techniques for data compression. To understand various channel coding techniques and their capability. To Build and understand the fundamental concepts of data communication and networking.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Data and Information, Communication Channels	2	Application on digital communication systems, advantages of encoding and decoding in communication systems
2	Basics of Information Theory: Entropy, Differential Entropy, Relative Entropy, Conditional Entropy, Mutual Information, Channel Capacity	6	Quantitative measure of information, fundamental theorems

3	Different Models of Sources, Channels and Noise: Discrete Source, Continuous Source, Memoryless Source, Memoryless Channel, Binary Symmetric Channel, Gaussian Channel, Additive White Gaussian Noise, Markov source	8	Knowledge of types of sources generating information
4	Source Coding and Channel Coding: Source Coding Theorem, Channel Coding Theorem	5	Types of encoding techniques for different sources
5	Quantum Information Theory: von Neumann Entropy, Quantum Relative Entropy, Quantum Joint Entropy, Quantum Conditional Entropy, Quantum Mutual Information	5	Introduction to quantum information and quantification
6	Coding Theory: Block Codes, Cyclic Codes, CRC Codes, BCH and Reed-Solomon Codes, Golay Codes, Convolutional Codes, Majority Logic Decoding, Viterbi Decoding Algorithm, LDPC Codes	12	Different coding techniques in binary space and higher order Galois field
7	STC coding: Introduction to SISO, MIMO, Space-time coding	4	Introduction to modern digital communication channels, channel matrix, Coding techniques
	Total	42	

Course Outcome	Perform information theoretic analysis of communication system. Design a data compression scheme using suitable source coding technique. Design a channel coding scheme for a communication system. Understand and apply fundamental principles of data communication and networking. Apply flow and error control techniques in communication networks. Apply Galois field coding techniques for communication networks
-----------------------	---

Learning Resources	1. Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, John Wiley & Sons, Inc., Second Edition, 2006. 2. Todd K. Moon, Error Correction Coding: Mathematical Methods and Algorithms, John Wiley & Sons, Inc., New Jersey, 2005. 3. Ron M. Roth, Introduction to Coding Theory, Cambridge University Press, 2006. 4. Modern Digital and Analog Communication Systems, B.P. Lathi , 5 th edition, Oxford University Press, USA. 5. Communication Systems, A. B. Carlson, 4th Edition Mcgraw Hill 2002 6. Ranjan Bose , Information Theory, Coding and Cryptography, TMH
---------------------------	---

Course Code	IT3124N	Course Name	Soft Computing Techniques	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Discrete mathematics, Calculus: Integration and Differentiation, Concepts of Algorithm and Programming</i>	Co-requisite Courses		Progressive Courses	<i>Machine Learning, Deep Learning</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	• To introduce the concepts and significance of soft computing in handling imprecision, uncertainty, and partial truth. • To understand and apply fuzzy logic principles to real-world problems. • To learn the fundamentals of neural networks and their applications in classification, pattern recognition, and function approximation. • To explore the principles of genetic algorithms and their role in optimization. • To integrate fuzzy logic, neural networks, and evolutionary computation in hybrid systems for solving complex engineering problems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Limitations of Artificial Intelligence; Definition of Soft Computing; Difference between Hard and Soft Computing; Domain soft computing techniques; Introduction to Fuzzy Systems, Artificial Neural Network, Genetic Algorithm, Hybrid Systems	2	Understand concepts and significance of soft computing in handling imprecision, uncertainty, and partial truth.

2	Fuzzy Logic System: Fuzzy Set Theory, Fuzzy Relation, Fuzzy Logic and Approximate Reasoning, Fuzzy logic system design, Applications. Advanced topics: Fuzzy clustering, Neuro-fuzzy system etc.	12	Learn fuzzy logic principles to design and solve real-world systems as well as problems.
3	Rough set theory: Introduction, Properties of Rough-set approximations, Attribute reduction. Rule generation, Rough set extensions, Application in data mining and Machine learning	8	Understands the principles of Rough set theory and its use in data mining and machine learning.
4	Artificial Neural Network (ANN): Basic electrical model of ANN, NN Architectures, Learning algorithms and paradigms, Learning Single layer and multilayer perceptions, Hopfield NN and Associative Memory, SOM Models and related algorithms, Applications	12	Understands the fundamentals of neural networks and their applications
5	Genetic Algorithm: Difference between Traditional Algorithms and GA, Encoding, Fitness Function, Reproduction, Cross Over, Mutation, Applications. Other Optimization techniques: PSO, Bio Inspired (ACO, SA, Bee Colony etc)	8	Grab the principles of genetic algorithms and their role in optimization.
	Total	42	

Course Outcome	At the end of the course, the student will be able to: ● CO1: Explain the fundamentals and importance of soft computing compared to conventional (hard) computing techniques. ● CO2: Apply fuzzy logic principles for modeling and solving uncertainty-based problems. ● CO3: Design and implement artificial neural networks for learning and classification tasks. ● CO4: Use of rough set theory in data mining and machine learning. ● CO5: Use genetic algorithms and other evolutionary techniques for solving optimization problems. ● CO6: Develop hybrid intelligent systems combining neural networks, fuzzy logic, and genetic algorithms. ● CO7: Analyze real-world problems using appropriate soft computing tools and techniques.
-----------------------	---

Learning Resources	TEXT BOOK: Neural networks A comprehensive foundations, Simon Haykin, Pearson Education 2nd Edition 2004 Neural Fuzzy Systems- A Neuro-Fuzzy Synergism to Intelligent System, C.T. Lin and George Lee, Prentice Hall Genetic Algorithms in Search, Optimization and Machine Learning, David E. Goldberg, Rough Sets: Theoretical Aspects of Reasoning about Data, Zdzisław Pawlak REFERENCE BOOKS: Artificial neural networks by B.Vegnanarayana Prentice Hall of India P Ltd 2005 Neural networks in Computer intelligence, Li Min Fu TMH 2003 Neural networks, James A Freeman David M S kapura Pearson Education An Introduction to Genetic Algorithms, Melanie Mitchell, MIT Press. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Jerry M. Mendel, Fuzzy Logic with Engineering Applications, Timothy. J. Ross Fuzzy Sets and Fuzzy Logic – Theory and Applications, G. J. Klir and Bo Yuan, Prentice Hall India
---------------------------	--

Course Code	IT3125N	Course Name	Graph Algorithms	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Basic knowledge on Algorithms, and Graph theory</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Introduction to fundamental graph concepts and provide a rigorous foundation for designing and analyzing graph-based algorithms. Problem-solving skills to be developed by applying graph traversal, shortest paths, spanning trees, flows, and matching algorithms to real-world problems.
-------------------------	---

	Students will know about the advanced graph-theoretic topics such as planarity, connectivity, partitioning, and graph optimization techniques. To impart knowledge about the analysis of algorithmic efficiency in terms of time and space complexity for graph problems. Usage of appropriate data structures to represent and manipulate graphs effectively in algorithmic implementations. To make aware about applications of graph algorithms in domains such as networks, bioinformatics, computer vision, and machine learning.
--	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Graph Traversal - BFS, DFS, Topological Sorting, Applications	3	Implement BFS and DFS algorithms for graph traversal. Apply BFS/DFS to solve problems like connectivity, cycle detection, and component identification. Perform topological sorting on directed acyclic graphs (DAGs).
2	Minimum Spanning Tree	4	Implement Kruskal's and Prim's algorithms for minimum spanning tree construction. Compare the performance of MST algorithms based on graph structure. Apply MST algorithms to solve real-world optimization problems (e.g., network design).
3	Shortest Paths	4	Apply all-pairs shortest path algorithms (Floyd–Warshall, Johnson's). Analyze algorithmic efficiency for different graph densities and edge weights.
4	Network Flows, Applications	4	Implement the Ford–Fulkerson method and Edmonds–Karp algorithm for maximum flow. Apply max-flow techniques to solve bipartite matching and circulation problems.
5	Matching on Bipartite Graph, Applications	4	Implement algorithms for maximum matching in bipartite graphs (e.g., Hopcroft–Karp). Apply matching algorithms to assignment and scheduling problems.

6	Eulerian and Hamiltonian Tours	4	Identify Eulerian and Hamiltonian graphs using theoretical criteria. Implement algorithms to find Eulerian circuits and paths.
7	Planar Graphs, Planarity Testing	5	Implement planarity testing algorithms (e.g., Kuratowski's theorem-based methods). Apply planar graph properties to VLSI design and geographic mapping problems.
8	Graph Partitioning and its Applications	5	Apply graph partitioning techniques to clustering and load balancing. Analyze the efficiency of partitioning algorithms.
9	Clique Partitioning	4	Implement algorithms for clique detection and partitioning. Apply clique partitioning to social network and bioinformatics applications.
10	Connected Components and its Applications	5	Implement algorithms to find connected components in undirected and directed graphs. Apply strongly connected components algorithms.
	Total	42	

Course Outcome	1. Implement and analyze fundamental graph algorithms for traversal, connectivity, and ordering, including BFS, DFS, and topological sorting. 2. Apply optimization techniques such as minimum spanning tree, shortest path, and network flow algorithms to solve real-world problems. 3. Design and implement algorithms for matching in bipartite graphs, Eulerian and Hamiltonian tours, and evaluate their computational complexity. 4. Apply planarity testing, graph partitioning, and clique partitioning techniques to practical scenarios such as clustering, load balancing, and VLSI design. 5. Identify and analyze connected components and strongly connected components in directed and undirected graphs, and apply these concepts to network analysis. 6. Compare alternative graph algorithms for efficiency, scalability, and applicability, justifying algorithm selection for specific problem contexts.
-----------------------	---

Learning Resources	Books: 1. Combinatorial Optimization, Theory and Algorithms (KV) by Bernhard Korte and Jen Vygen, Springer, 4th Edition (2008). (GMU's libraries have online versions of this book.) 2. Networks, Crowds, and Markets: Reasoning about a Highly Connected World by David Easley and Jon Kleinberg, Cambridge University Press, (2010). 3. Introduction to Graph Theory by Douglas B. West, 2nd Edition (2000). 4. Combinatorial Optimization: Algorithms and Complexity by C. H. Papadimitriou and K. Steiglitz, Englewood Cliffs, Prentice Hall, c1982, Reprinted by Dover Books, (1998). 5. Algorithm Design (KT) by Jon Kleinberg and Eva Tardos, Pearson Education, Inc. (2006). 6. Introduction to Algorithms by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, The McGraw-Hill Companies, 2nd Edition (2001). 7. Algorithmic Graph Theory and Perfect Graphs by Martin Charles Golumbic, North Holland, 2nd Edition (2004).
---------------------------	--

Course Code	IT3126N	Course Name	Digital Signal Processing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Signals, systems and circuits, Communication systems</i>	Co-requisite Courses	<i>Microprocessor and Microcontroller</i>	Progressive Courses	<i>Wireless Networks, Information and Systems Security</i>
Course Offering Department	<i>Information Technology</i>		Data Book / Codes/Standards		

Course Objective	1. Understanding Discrete-Time Signals and Systems 2. Learning the Z-transform 3. Designing FIR and IIR filters to meet specific frequency-domain requirements 4. Understanding DFT and FFT for time and frequency domain analysis of discrete signals 5. Different filter design and noise analysis
------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Discrete time Signals and Systems- Time and Frequency domain representation	2	Signal representation
2	Discrete Signals: Z-Transforms, inverse z-Transform, properties of Z-transform. Difference equations and solution.	2	Mathematical representation of discrete systems, discrete signal time

			and frequency domain analysis
3	Linear time-invariant system and their properties, Stability, Frequency Response; Linear phase systems	4	Discrete time systems and analysis
4	Realization of Digital Filters: Recursive and non recursive structures. IIR Filters- Block Diagrams and signal flow graphs, direct cascade, parallel, ladder and lattice realizations	4	Types of digital filters and their time and frequency domain analysis
5	Realization of Digital Filters: FIR Filters, Lattice structure. Quantization Effect	6	Types of digital filters and their time and frequency domain analysis
6	Digital Filter Design: IIR Filter- Approximation theory, Impulse invariant and bilinear transformations, Frequency transformations.	2	Types of digital filters and their time and frequency domain analysis
6	FIR Filter Design: Windows and Frequency sampling techniques.	4	Truncation methods of digital filters
7	Discrete Fourier Transforms- Definitions, and properties, Circular convolution, Linear convolution.	6	Frequency domain analysis of discrete point set
8	FFT Algorithms- Basic DIT and DIF algorithms, Computational efficiency considerations.	4	Improvement over DFT
9	Finite Word Length Effects- Quantization error and their effects on performance of digital signal processors	4	Quantization noise reduction
10	Multirate filter design- Decimation, application	4	Modern DSP based filter design and application
	Total	42	

Course Outcome	Students will learn about analysis of discrete time signals, Discrete Fourier Transform and digital filter design and quantization effect
-----------------------	---

Learning Resources	Books: 1. John G. Proakis (Author), Dimitris G. Manolakis, "Digital Signal Processing: Principles, Algorithms, and Applications", 4e, Pearson 2. A. H. Oppenheim, R. W. Schaffer, "Digital Signal Processing", 2015, Pearson 3. Sanjit Mitra, "Digital Signal Processing", 4th ed.
---------------------------	---

Course Code	IT3127N	Course Name	Design Practices	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	This course is designed to make students aware of the approach, techniques, breadth of System Design space given a set of functional and non-functional requirements. Looking at the prevalent trend of system design interviews at top organizations worldwide, a special attention is given to make students delve deep into applications of everyday use putting on the cap of a system designer.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	System design basics : Basic understanding of functional and non-functional requirements, related artifacts they go in, families of applications - thick client, web applications, single page applications, API, business intelligence. Use case diagrams, entity relationship diagrams, sequence diagrams, package dependencies	3	Knowing about functional requirements and application families
2	Non-Functional Requirements: Understanding reliability, availability, performance, throughput, scalability, portability, maintainability of applications and how they impact system design trade-offs - latency vs throughput, performance vs scalability, consistency vs availability, CAP theorem	4	How NFR and trade-offs influence system design decisions
3	Storage types: Relational and Non-Relational Storages, ACID transactions, eventual consistency, basics of data quality checks, basics of metadata	4	Tour of storage systems and databases
4	Security Considerations: Authentication vs Authorization, Issues around simple login, SSO, OAuth2 and OpenIDConnect, Maintaining Keys and secrets, Encryption of data in motion and rest, Certificates, Identity Federation	4	Knowing about Web, API, storage security considerations
5	Integration design: File based, API based, database oriented, event based, batch systems, Communication - synchronous & asynchronous,	4	Understanding types of integration among applications

			and data exchange
6	Distributed Systems: Consistency, Replication, Distributed Transactions, Orchestrations, , exploratory analytics, End User Computing High Availability & Disaster Recovery	5	Review of distributed systems wrt NFRs
7	Types of applications: Monolith, Microservices, Modular monolith, CQRS – pros & cons, scenarios when they are used	4	Application architecture types – when to use what
8	Techniques in System Design: Load balancing, Caching, Partitioning of data, Creating data models for OLTP, Data model for Data Warehouse , fundamentals of batch and streaming ETLs	4	Techniques used in real-life applications in Production
9	Cloud Service Contexts: Azure (could be AWS) Blob Storage, App Services, Key Vault, Auto-scaling, Azure Container Service, Virtual Machines, CosmosDB, Azure SQL Database, Event Grid, Service Bus, AI services in Azure/AWS - familiarity	5	Applying the learnings above in public cloud landing zones
10	Pictorial view of tools, services, vendor systems used in some of the most successful applications known to mankind Case Studies: Designing a Social Network System Case Studies: Designing a Video Streaming Service Case Studies: Designing an Airbnb type application	5	Case studies and knowing about the major providers in the eco-system
	Total	42	

Course Outcome	• Students will learn to analyze requirements from functional and non-functional angles • Students will learn to take storage, services, integration aspects while designing an application, including distributed ones • Students will learn how to apply the principles of system design on a public cloud provider's landing zone • Students will gain confidence in attending various system design interviews by FAANG companies
-----------------------	--

Learning Resources	1. System Analysis and Design with UML – Allan Denis, Wiley Publication 2. System Design on AWS – Jayanth Kumar, Oreilly Publications 3. Acing system design interview – Zhiyong Tan, Manning Publications
---------------------------	--

Course Code	IT3128N	Course Name	Computer Graphics	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Computer Programming, Data Structure and Algorithms</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To enable students to learn salient topics of Computer Graphics and their applications.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Computer Graphics: Overview, applications and software, basic graphics I/O devices, overview of Raster and vector graphics display working principle of CRT based display device, LCD display device. Introduction to frame buffer, Colour Look Up Table etc.	6	Students will learn basics of Computer Graphics
2	Scan Conversion: Scan Converting Lines: DDA, Bresenham, Mid-point algorithms and Problems of Aliasing Scan Converting Circles and Ellipse	7	Students will learn scan-conversion algorithms
3	Filling Polygons: Flood fill, boundary fill, scan-line fill	4	Students will learn polygon-filling algorithms

4	Clipping Algorithms: Line clipping algorithms: Cyrus-Beck, Cohen-Sutherland, Liang-Barsky Polygon Clipping algorithms: Sutherland-Hodgman and Weiler-Artherton algorithm	3	Students will learn clipping algorithms
5	Two-Dimensional Transformations: Transformations and Matrices Transformation Conventions Basic 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Combined Transformations, Window-to-Viewport Transformations	7	Students will learn algorithms for two-dimensional transformations
6	Three-Dimensional Transformations and Projections: Introduction, Basic transformation matrices in Three- Dimensional Space , Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Projections: Orthogonal, axonometric, and oblique	7	Students will learn algorithms for three-dimensional transformations and projections
7	Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms , Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division methods: BSP trees, Visible-Surface Ray Tracing	5	Students will learn algorithms for visible-surface determination
8	Plane Curves and Surfaces: Curve Representation, Representation of Space Curves: Cubic Splines, Bezier Curves, B-spline Curves	3	Students will learn plane curves and surfaces
	Total	42	

Course Outcome	On successful completion of this course, students will learn: 1. Basics of Computer Graphics 2. Scan-conversion algorithms 3. Polygon-filling algorithms 4. Clipping algorithms 5. Algorithms for two-dimensional transformations 6. Algorithms for three-dimensional transformations and projections 7. Algorithms for visible-surface determination 8. Plane curves and surfaces
-----------------------	--

Learning Resources	1. Computer Graphics with OpenGL (3/e), D. D. Hearn and M. P. Baker. 2. Computer Graphics Principles & Practice by James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, 2nd Edition in C. 3. Mathematical Elements for Computer Graphics by Rogers and Adams, McGraw Hill. 4. Computer Graphics (First Indian Edition), Peter Shirley and Steve Marschner, Cengage Learning Reprint of A. K. Peters, 2011.
---------------------------	--

Course Code	IT3129N	Course Name	Compiler Design	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Formal language and Automata Theory</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>		Data Book / Codes/Standards		

Course Objective	Understand the principles and phases of a compiler, Apply parsing techniques to syntax analysis, Introduce optimization techniques and runtime environments, Design and implement simple compilers or interpreters
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: High level and low level language, Compiler, types, challenges in compiler design, phases and passes, language processing system,	2	Basic of Compiler
2	Lexical analyzer, role, issues, Token , pattern, lexeme, lexical errors, input buffering, Regular expression, regular definitions, recognition of tokens, transitional system, Conversion of RE to NFA-Thomson's construction, Converting RE directly to DFA, Minimizing DFA, Lex compiler	6	Knowledge of Lexical Analyzer
3	Formal grammars, and their application to syntax analysis, BNF notation, ambiguity, Removal of left recursion and left factoring Top down parsing, Recursive decent parsing, Predictive parsing, Recursive predictive parsing, Non recursive predictive parsing, LL(1) parsing, First, Follow, LL(1) table constructing, not LL(1), Error recovery in predictive parsing-panic mode and	14	Different parsing techniques

	phrase level Bottom up parsing, handle, shift reduce parsing, problems/ conflict in shift reduce parsing, operator precedence parsing, LR parsing, SLR, SLRtable constructing, canonical and LALR, YACC		
4	Syntax directed transition, syntax directed definition (SDD), Attribute grammar, SDD for type checking, Abstract syntax tree, synthesized and inherited attribute, dependency graph, S and L attribute, Semantic error	6	Knowledge of SDD and SDT
5	Type checking, Static and dynamic check, type system, type expression, error recovery, specification of a simple type checker,	4	Knowledge of type checking
6	Intermediate code generation-advantages, forms of intermediate representation (Syntax tree, DAG, Three address code), Three address code(3AC)-types, quadruples, triple and indirect triple, 3AC for relation and logical statement, Boolean expression, condition statement and loop, Backpatching	4	Knowledge of Intermediate code generation
7	Code generation: factor affecting code generation, register allocation, basic block and flow graph, transformation on basic blocks-structure preserving, common sub expression elimination, dead code elimination, renaming temporary variable, algebraic transformation, DAG and basic block, peephole optimization,	3	Knowledge of code optimization
8	Run time environment: Memory management, Storage organization, static vs dynamic allocation, activation tree, control stack, storage organization, heap allocation, activation records, garbage collection- mark and sweep , reference counting, generation garbage collection, 3 partial garbage collector.	3	Knowledge of runtime environment and memory management
	Total	42	

Course Outcome	After going through the subject students will be able to 1. Explain the phases of a compiler and the underlying principles of lexical, syntax, semantic, and intermediate code analysis. 2. Apply parsing and syntax analysis techniques to construct and validate grammar for programming languages. 3. Design and implement intermediate code generation, type checking, code optimization, and error handling. 4. Develop efficient code generation and runtime environment management including memory allocation and garbage collection.
-----------------------	---

Learning Resources	1. A. V. Aho, R. Sethi and J. Ullman, Compilers: Principles, Techniques and Tools, Addison Wesley. 2. Steven S. Muchnick, Advanced Compiler Design and Implementation, Morgan Kaufmann Publishers. 3. David Galles, Modern Compiler Design, Pearson Education. 4. C. Fischer and R. LeBlanc, Crafting a Compiler, Benjamin Cummings. 5. A. Appel, Modern Compiler Implementation in C, Cambridge Univ. Press, London. 6. C. Fischer and R. LeBlanc, Crafting a Compiler in C, Benjamin Cummings.
---------------------------	---

Course Code	IT3130N	Course Name	Fundamentals of Information Retrieval	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	The course is to understand the fundamentals of Information Retrieval and the science behind search engines. The principles mentioned here will enable students to apply it to large varieties of systems including linguistics, medical informatics, legal systems. The course is also a stepping stone for Machine Learning and NLP based systems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Boolean Retrieval Term Vocabulary and posting lists: Document delineation and character sequence decoding, determining the vocabulary of terms, skip pointers	4	Inverted indexes, boolean query processing, document pre-processing

2	Dictionaries & tolerant retrieval: Search structures for dictionaries, Wildcard queries, spelling correction, phonetic correction Index construction: Blocked-sort based indexing, single-pass in-memory indexing, dynamic indexing,	5	Search structure of dictionaries, matching documents with imprecise inputs, algorithms for building inverted index from text collections,
3	Index compression Scoring, term weighting, and vector space model: parametric & zone indexes, term frequency & weighting, vector space model for scoring, variant tf-idf functions, feedback mechanisms	4	Non-boolean retrievals, tf-idf
4	Computing scores in a complete search system: efficient scoring & ranking - cluster pruning, impact ordering, components of an IR system - tiered indexes, query term proximity, IR system evaluation, relevance feedback and query expansion	5	Listing documents rank ordered for queries, evaluation of IR systems
5	Language Models for IR: Language models, Query likelihood model Text Classification & Naive Bayes: Naive Bayes text classification, Bernoulli model, Feature selection	5	Applying probability theory on IR
6	Vector space classification: Document representations & measures of relatedness, k nearest neighbor, linear vs non-linear classifiers, bias-variance trade-offs	4	Document vectors, vector space models, kNN classifications
7	Support Vector Machines & ML on documents: SVM, Extensions of SVM, Flat clustering, hierarchical clustering, matrix decomposition & latent semantic indexing	5	SVM systems and its effectiveness for text classification, building clusters from related documents
8	XML Retrieval, Probabilistic Information retrieval - binary independence model, IR from Geographic Information Systems,	5	Retrievals from various systems e.g. XML, GIS

	Web Search basics, Web crawling & indexes, link analysis - page rank		
9	Management of IR systems: Knowledge management, digital assets management, digital content licensing, SEO	5	Managing large IR systems
	Total	42	

Course Outcome	- Students will know the methods used in finding materials of interest from unstructured corpus - Students will know about the techniques used for boolean retrievals and rank based retrievals - Students will know various practical techniques used in indexes facilitating retrieval from unstructured corpus such as texts - Students should be able to relate to the topics of machine learning and preparing data from corpus to feed various models - Lastly students will know about retrievals from XML documents, GIS and governance methods used in large IR systems
-----------------------	--

Learning Resources	- An introduction to Information Retrieval – Christopher Manning , Cambridge university press - Understanding Information Retrieval Systems – Marcia Bates, CRC Press
---------------------------	--

Course Code	IT3161N	Course Name	Image Processing	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Signals and Systems, Discrete Math</i>	Co-requisite Courses	<i>Digital Signal Processing</i>	Progressive Courses		Computer Vision	
Course Offering Department		Information Technology		Data Book / Codes/Standards			

Course Objective	Image Processing is the core of visual information processing with promise and potential of its diverse domain applications ranging from computer vision problem, industrial automation, medical disease diagnosis and prediction, weather forecasting etc. The objective of the course is to make the students conversant with digital image formation, imaging, image quality improvement, restoration, object-background partition, shape analysis and efficient storage and transmission.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction: Image definitions. Image representation: monochrome and color models, image file formats; image digitization, sampling and quantization, image resolution. Image Enhancement & Feature Extraction, Image Analysis and pattern Recognition	4	Introduction to digital image, formation, various image processing operations and applications
2.	Image processing tools: Fourier, Hadamard-Walsh, Discrete cosine, wavelets and multiresolution analysis; mathematical morphology - binary morphology, dilation, erosion, opening and closing, duality relations	6	Familiarization of different image processing tools, merits and limitations
3.	Image Enhancement: Filters in spatial and frequency domains, histogram- based approaches, smoothing, edge enhancement and image sharpening filtering, Homomorphic filtering.	8	To improve image quality improvement, different techniques-spatial and frequency domain approaches
4.	Image Restoration - PSF, circulant and block - circulant matrices, deconvolution, restoration using inverse filtering, Wiener filtering and constrained least square method.	6	Inverse problem: Modeling of image degradation and restoration
5.	Segmentation: Pixel classification, Bi-level thresholding, multilevel thresholding, split and merge algorithm, region growing, texture and entropy- based methods.	6	Partitioning of Object and Background-techniques
6.	Image compression: requirements and types, Statistical and spatial compression techniques like RLE, PCM, Huffman coding etc. Transform coding algorithm-DCT, Concept of Hybrid coding, (Quantization and sampling, Uniform quantizer and non uniform quantizer, Concept of JPEG and MPEG standards)	4	Efficient storage and transmission of image compression-lossy and lossless techniques
7	Color Image Processing: Color Representation, Laws of color matching, chromaticity diagram, color enhancement, color image segmentation, color edge detection	4	Introduction to color image, enhancement and segmentation
8	Image Processing Applications: Weather, Agriculture, Disease diagnosis, Robot Automation, Mining and Material Science	4	Use Image Processing knowledge on diverse applications
	Total	42	

Course Outcome	CO1: To understand digital image formation, fundamental of image processing operation CO2: To understand image quality improvement, object-background partitioning CO3: To understand image restoration from degraded versions CO4: To understand efficient storage and transmission of Images CO5: Use of Image Processing knowledge for diverse applications
-----------------------	--

Learning Resources	Books: 1.Digital Image Processing: Rafael Gonzalez and Richard E. Woods, Pearson 2.Digital Image Processing and Analysis: Chanda and Dutta Majumder, PHI 3.Digital Image Processing: A practical introduction using JAVA: Nick Efford, Pearson Education 4.
---------------------------	---

Course Code	IT3171N	Course Name	Microprocessor and Microcontroller Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Digital Logic and Circuit Design</i>	Co-requisite Courses	<i>Programming and Algorithms</i>	Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	CO1. Provide hands-on experience with microprocessor and microcontroller architecture. CO2. Programming in assembly language, and CO3. Interfacing with peripheral devices to solve real-world problems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Study of programs on trainer kit (8085) using the basic instruction set (data transfer, Load/Store, Arithmetic, Logical). Assignments based on theory subject.	2	Learn to use 8085 kit and programming on it
2	Familiarization with 8085 simulator on PC. Study of programs using basic instruction set (data transfer, Load/Store, Arithmetic, Logical) on the simulator. Assignments based on above.	4	Learn assembler of 8085

			with instruction set
3	Program using subroutine calls and IN/OUT instructions using 8255 PPI on the trainer kit eg, subroutine for delay, reading switch state & glowing LEDs accordingly, finding out the frequency of a pulse train etc.	6	Learn I/O interfacing
4	Interfacing any 8-bit Latch (e.g., 74LS373) with trainer kit as a peripheral mapped output port with absolute address decoding	4	Learn I/O interfacing and address mapping
5	Interfacing with I/O modules: ADC, Speed control of mini DC motor using DAC, Keyboard, Multi-digit Display with multiplexing, Stepper motor.	6	Learning other I/O interfacing applications
6	Study of programs on 8086 trainer kit using the basic instruction set (data transfer, Load/Store, Arithmetic, Logical). Assignments based on theory subject.	6	Use of 8086 kit and doing program on it
7	Familiarization with 8086 simulator on PC. Study of programs using basic instruction set (data transfer, Load/Store, Arithmetic, Logical) on the simulator. Assignments based on above.	4	Learn assembler of 8086 with instruction set
8	Assignments based on Assembly Language programming of 8051.	6	Writing program on 8051
9	ARM MBED based Application Development.	4	Learn how to develop ARM microcontroller based embedded applications
	Total	42	

Course Outcome	1. Write and Execute Assembly level programs for 8085, 8086 and 8051 2. Design and implement circuits to interface microprocessors and microcontrollers with external peripherals. 3. Debug and troubleshoot assembly language programs and hardware interfaces
-----------------------	---

Learning Resources	1. Ramesh S. Goankar, "Microprocessor Architecture, Programming and Applications with 8085", 5th Edition, Prentice Hall India 2. 8051 Microcontrollers by Mazidi 3. Microprocessor and Interfacing by Douglas V. Hall
---------------------------	---

Course Code	IT3172N	Course Name	Database Management Systems Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	- Modeling of Database - Learning Database Query Languages
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	ER Modelling	6	Understanding Entity-Relationships
2	Creation of Tables using SQL, Integrity Constraints in SQL	6	Table creation with key constraints
3	SQL DDL statements and commands	6	Data Definition Language
4	Execution of DML statements and queries in SQL	6	Data Manipulation using SQL
5	User privileges and grants	3	Access grants
3	PL/SQL Programming for Small Application	6	Procedural SQL
4	Programming using Function, Procedure, Cursor and Trigger, MySQL	6	Complex programming using SQL

5	SQL Application Programming, No SQL (Redis), Neo4j	3	Application development
	Total	42	

Course Outcome	Learn to develop database models and queries for database access
-----------------------	--

Learning Resources	1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, "Database System Concepts", Mc Graw Hill, 6th ed, 2013. 2. Ramez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson, 7th ed, 2016. 3. C. J Date, "An Introduction to Database System", Pearson, 8th ed, 2003. 4. Ivan Bayross, "SQL, PL/SQL: The Programming Language of Oracle" BPB Publications, 4th ed, 2010
---------------------------	--

Course Code	IT3173N	Course Name	Operating Systems Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses		Co-requisite Courses	<i>Operating Systems (IT3103N)</i>	Progressive Courses	
Course Offering Department	<i>Information Technology</i>		Data Book / Codes/Standards		

Course Objective	To get hands-on understanding of the concepts taught in the Operating Systems theory course (IT3103N).
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Understanding the time sharing, multiprogramming nature of the operating system	6	Simulate time sharing, multiprogrammin

			g environm ent
2	Inter-process communication using shared memory and message passing	6	Simulate inter-process communication
3	Thread related programs: scheduling of threads, master-slave model	6	Simulate thread related programs
4	Simulation of different scheduling algorithms	6	Simulate scheduling algorithms
5	Solving different classical problems of synchronization using semaphores and monitors	6	Simulate process synchronization
6	Simulation of different detection, prevention and avoidance algorithms for deadlocks	6	Simulate algorithms for deadlocks
7	Analyzing the cache/memory behavior of systems (memory mountain etc.), File operations	6	Simulate memory and file operations
	Total	42	

Course Outcome	On successful completion of this lab, students will get hands-on understanding of the concepts taught in the Operating Systems theory course.
-----------------------	---

Learning Resources	Same as the Operating Systems theory course (IT3103N).
---------------------------	--

Sixth Semester

Course Code	IT3201N	Course Name	Computer Networks	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Basic knowledge on Algorithms, and Graph theory, Operating Systems</i>	Co-requisite Courses		Progressive Courses	<i>Advanced computer Network, Network security and cryptography, Cloud computing, Wireless communication and mobile Network, Software defined network</i>
Course Offering Department	Information Technology			Data Book / Codes/Standards	

Course Objective	• To introduce fundamental concepts, models, and architectures of computer networking. • To explain the layered approach to networking with emphasis on the OSI and TCP/IP models. • To develop understanding of network protocols and communication standards. • To explore the principles behind routing, switching, and congestion control. • To enable students to analyze network performance and security issues.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Computer networks and their types; a brief history of networking; data switching techniques; an introduction to virtual circuit switched networks and datagram networks; need of layered architecture; ISO-OSI and TCP/IP architecture; task of layers	6	Learn layered architecture of ISO-OSI layer and their functionalities
2	Transmission media; bit rate, baud rate and bandwidth; bit encoding techniques; Network topology	6	Understands basic terminologies related to

			data communication
3	Data link layer (DLL) design issues; Error detection and correction; Flow control; Example DLL protocols – HDLC and PPP; Need of MAC sub-layer; ALOHA and CSMA protocols; Ethernet LAN; HUB, bridge and switch; Switched LAN; an introduction to Token ring LAN and FDDI; DQDB; Wireless LAN	12	Will Learn in detail about DLL layers and its protocols
4	Need of network layer; Routing algorithms and protocols – RIP, OSPF, BGP; router; routing in Internet; need of logical address; X.25 network, ARPANET and ERNET;	8	Learn in detail about addressing mechanisms, subnetting concepts, routing algorithms etc
5	internetworking; network layer in Internet – IP, ICMP, IGMP, ARP, RARP, DHCP; NAT and CIDR; IPv6	10	Learn working of different IP protocols
6	Design issues of transport layer; socket address; congestion control; TCP and UDP	10	Learn protocols of TCP and UDP
7	Introduction to application layer protocols; SMTP, FTP, HTTP; network management and security	4	Learn different application layer protocols
	Total	56	

Course Outcome	By the end of the course, students will be able to: ● CO1: Describe the functions of each layer in the OSI and TCP/IP models. ● CO2: Analyze and explain the working of common network protocols such as HTTP, FTP, TCP, UDP, IP, ARP, and ICMP. ● CO3: Compare and contrast different routing algorithms and switching techniques, IP Protocols, Congestion control algorithms. ● CO4: Demonstrate understanding of addressing schemes (IPv4, subnetting, CIDR) and data encapsulation. ● CO5: Evaluate network performance using metrics such as latency, throughput, and jitter. ● CO6: Identify security threats and apply basic security techniques (firewalls, encryption).
-----------------------	---

Learning Resources	1. L. L. Paterson and B. S. Davie: Computer Network, Morgan Kaufman, San Mateo. 2. A. Tannenbaum: Computer Networks, Prentice Hall India. 3. W. Stallings: ISDN and Broadband ISDN With Frame Relay and ATM, Prentice Hall. 4. W. Stallings: Local and Metropolitan Area Networks, Macmillan, New York 5. Kaufman, R. Perlman and M. Speciner: Network Security, Prentice Hall, Englewood Cliffs 6. V. P. Ahuja: Design and Analysis of Computer Communication Networks, McGraw Hill, New York 7. L. Gracial and I. Widjaja: Communication Networks, Tata-McGraw Hill. 8. L. L. Paterson and B. S. Davie: Computer Network, Morgan Kaufman, San Mateo.
---------------------------	---

Course Code	IT3202N	Course Name	Machine Learning	Course Category	PC	L	T	P
						3	1	0

Pre-requisite Courses	<i>Linear Algebra, Probability, Statistics</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Introduce the fundamental concepts of machine learning, including well-posed learning problems, supervised, unsupervised, and reinforcement learning. Impart mathematical and algorithmic foundations for various learning paradigms such as regression, classification, clustering, and probabilistic learning. Enable to design, implement, and evaluate machine learning algorithms using appropriate data structures, optimization techniques, and evaluation metrics. Inculcate analytical skills to compare and select suitable learning models and approaches based on the nature of data and problem requirements. Exposure to modern techniques such as neural networks, ensemble learning, and regularization to address complex real-world problems. Encouraging application and innovation by applying ML techniques in practical domains like computer vision, natural language processing, healthcare, and business intelligence.
-------------------------	---

Module	Module Name and Topics	Duration (class-hour)	Module Outcome
---------------	-------------------------------	------------------------------	-----------------------

1	Introduction Learning Problems, Well-posed learning problems, Designing learning systems.	4	Identify the characteristics of well-posed learning problems. Outline the key components in designing an effective learning system.
2	Concept Learning Concept learning task, Inductive hypothesis, Ordering of Hypothesis, General-to-specific ordering of hypotheses. Version spaces, Inductive Bias.	4	Formulate a concept learning problem and describe the concept learning task. Explain inductive hypothesis formation and hypothesis space ordering.
3	Learning Rule Sets Sequential Covering Algorithm, First Order Rules, Induction, First Order Resolution, Inverting Resolution.	4	Implement the sequential covering algorithm for rule learning. Apply first-order logic to represent and learn rules. Use resolution and inverted resolution in inductive logic programming.
4	Regression Linear regression, Notion of cost function, Logistic Regression, Cost function for logistic regression, application of logistic regression to multi class classification.	6	Apply logistic regression for binary classification tasks. Extend logistic regression to multi-class classification using one-vs-rest or softmax methods. Analyze regression model performance using error metrics.
5	Supervised Learning Support Vector Machine, Decision tree Learning, Representation, Problems, Decision Tree Learning Algorithm, Attributes, Inductive Bias, Overfitting. Bayes Theorem, Bayesian learning, Maximum likelihood, Least squared error hypothesis, Gradient Search, Naive Bayes classifier, Bayesian Network, Expectation Maximization Algorithm.	10	Implement support vector machines for classification tasks. Construct decision trees using appropriate splitting attributes and pruning methods. Apply Bayes theorem for probabilistic learning. Implement Naive Bayes classifiers for classification problems. Apply the Expectation–Maximization (EM) algorithm for parameter estimation.

			Analyze the effect of inductive bias and overfitting in supervised models.
6	Unsupervised learning Clustering, K-means clustering, hierarchical clustering.	6	Implement K-means clustering and evaluate cluster quality. Apply hierarchical clustering to discover group structures in data. Compare clustering algorithms based on performance and scalability.
7	Instance-Based Learning k-Nearest Neighbour Algorithm, Radial Basis Function, Locally Weighted Regression, Locally Weighted Function.	8	Implement the k-nearest neighbour algorithm for classification and regression. Apply radial basis function networks for function approximation. Implement locally weighted regression and analyze its performance.
8	Neural networks Linear threshold units, Perceptrons, Multilayer networks and back propagation, recurrent networks. Probabilistic Machine Learning, Maximum Likelihood Estimation.	8	Implement perceptrons and multilayer feedforward networks with backpropagation. Apply recurrent neural networks to sequence data. Use probabilistic machine learning concepts to model uncertainty in predictions.
9	Regularization, Preventing Overfitting, Ensemble Learning: Bagging and Boosting, Dimensionality reduction	6	Apply regularization techniques such as L1 and L2 to control overfitting. Implement ensemble methods like bagging and boosting to improve model accuracy.
	Total	56	

Course Outcome	1. Explain fundamental concepts of machine learning, types of learning problems, and the design principles of learning systems.
-----------------------	--

	2. Apply concept learning, inductive hypothesis formulation, version spaces, and rule-based learning methods to classification tasks. 3. Implement and evaluate regression models, logistic regression, and supervised learning algorithms such as SVM, decision trees, Bayesian classifiers, and the Expectation–Maximization algorithm. 4. Apply unsupervised learning methods, including K-means and hierarchical clustering, to discover patterns and group structures in data. 5. Implement instance-based learning methods such as k-NN, RBF networks, and locally weighted regression, and compare them to model-based approaches. 6. Design and train neural networks, including perceptrons, multilayer feedforward networks, recurrent networks, and probabilistic models, for real-world application
--	--

Learning Resources	1. Machine Learning, Tom Mitchell, McGraw Hill, 1997. 2. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2e), Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer Science and Business Media, 2009. 3. Pattern Classification (2e), Richard o. Duda, Peter E. Hart, David G. Stork, John Wiley & Sons, 2012. 4. James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning. Vol. 112. New York: springer, 2013. 5. Christopher M. Bishop. Machine learning and pattern recognition. Information science and statistics. Springer, Heidelberg. 2006. 6. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.
---------------------------	--

Course Code	IT3221N	Course Name	Artificial Intelligence	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Algorithms, Discrete Maths</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	Understand the fundamentals of Artificial Intelligence Develop an understanding of where and how AI can be used Develop proficiency in various AI problem-solving techniques
------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Overview of AI, agents & environment, Nature of environment, Different types of agents.	3	Introduction to Ai, agent
2	Problem solving: Problem, State space, Solving problems by searching, Uninformed and informed search, Breadth first search, Depth first search, Bi- directional search, Iterative deepening search	5	Understand the different Problem solving approaches
3	Heuristic search: Hill climbing, Best first search, Branch and bound, A* algorithm, Admissibility and monotonicity of A*, Iterative deepening A*, Simulated Annealing, Constraint satisfaction problem	8	Learn the informed search
4	Adversarial search: Games, Optimal decisions & strategies in games, Minimax algorithm, Alpha-Beta pruning.	3	Understand Adversarial search, Minimax algorithm
5	Knowledge and Reasoning: Knowledge representation issues, Predicate Logic, First order logic and WFFs, Forward reasoning, Unification, Resolution refutation in FOL, Backward reasoning, Structured knowledge representation, Semantic networks, Frames.	7	Understand Knowledge representation, Predicate Logic, Reasoning:
6	Introduction to logic programming : Basic knowledge of Prolog programming	4	Learn logic programming using Prolog
7	Probabilistic reasoning: Representing knowledge in an uncertain domain, the semantics of Bayesian networks, Shafer theory	4	Represent knowledge in an uncertain domain
8	Planning: Components of a planning system, Forward and backward state space planning, Goal stack planning, Hierarchical planning.	4	Represent components of a planning system
9	Learning: Concept of learning , learning by induction , explanation-based Learning, introduction of neural networks.	2	Understand the concept of learning
10	Expert Systems: Expert system architectures, Expert system shells knowledge acquisition.	2	Understand the basic of Expert Systems
	Total	42	

Course Outcome	1. Explain the fundamental concepts and scope of AI. 2. Apply various problem-solving strategies (uninformed and informed search) to AI problems. 3. Design and implement knowledge representation schemes and reasoning techniques.
-----------------------	--

	- Analyze problems using logic programming, constraint satisfaction, and probabilistic models. - Demonstrate the working of AI algorithms in areas like natural language processing, planning, and expert systems.
--	---

Learning Resources	Suggested Readings: - Ritch & Knight, Artificial Intelligence, TMH - Stuart Russel Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson - Patterson, . Introduction to Artificial Intelligence & Expert Systems, PHI N. J. Nilsson. Artificial Intelligence : A New Synthesis, Elsevier India - Ivan Bratko, PRPLOG: Programming for Artificial Intelligence, Pearson
---------------------------	---

Course Code	IT3222N	Course Name	Real Time Systems	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Basic knowledge on OS and networks.</i>	Co-requisite Courses	<i>Embedded system design</i>	Progressive Courses	<i>Cyber physical systems, Safety critical system design</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	- To introduce the fundamental concepts and characteristics of real-time systems. - To understand the hardware and software components of real-time embedded systems. - To explore scheduling algorithms and timing analysis for real-time task management. - To study real-time operating systems (RTOS) and their services. - To enable students to design and analyze real-time applications with timing and reliability constraints.
------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Introduction to the idea of real time and real time systems. Examples: Real time applications, Hard vs. soft real time systems.	2	Will learn characteristics, types, and applications of real-time systems.
2	Reference model of real time systems: Processors and resources, Tasks and their timing parameters, precedence constraints and data dependency	6	Analyze the architecture and timing constraints in real-time embedded systems

3	Real Time scheduling: Different Approaches- Clock Driven, Priority Driven, Scheduling of Periodic and Sporadic Jobs in Priority- Driven Systems.	9	Learn different RT scheduling algorithms
4	Resource access Management: Resources and Resource Access Control, Critical Section, Priority-Ceiling Protocols, concurrent Access to Data Objects.	9	Understand different RT scheduling algorithms with resource constraints
5	Real Time OS: RTOS Overview, RTOS Components, Task Management & Memory Management, Intertask Communication and Synchronisation, Kernels, Commercial Real-time Operating Systems.	8	Understand about RTOS and embedded platforms.
6	Real time communications: Communication medium, Real time traffic, synchronous and asynchronous message, performance comparison, real time communication protocols.	8	Learn real time communication aspects and its protocols
	Total	42	

Course Outcome	At the end of the course, the student will be able to: • CO1: Explain the characteristics, types, and applications of real-time systems. • CO2: Analyze the architecture and timing constraints in real-time embedded systems. • CO3: Apply scheduling algorithms such as Rate Monotonic, Earliest Deadline First, etc., for real-time task management. • CO4: Demonstrate the use of real-time operating system (RTOS) concepts like task scheduling, semaphores, and inter-process communication. • CO5: Develop and simulate real-time applications using RTOS and embedded platforms. • CO6: Evaluate the performance and reliability of real-time systems under timing constraints.
-----------------------	--

Learning Resources	Books: 1. Real Time Systems by Jane W. S. Liu, Pearson Education 2. Real Time Systems by C. M. Krishna and Kang G. Shin, McGrawHill Education 3. Real Time Systems: Theory and Practice, Rajib Mall, Pearson Education 4. Embedded Systems: Architecture, Programming and Design by Raj Kamal, Tata McGraw Hill Education
---------------------------	--

Course Code	IT3223N	Course Name	Embedded Systems and IoT	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic and Circuit Design, Computer Organization and Architecture, Microprocessors and microcontrollers, Basic programming concepts	Co-requisite Courses	<i>Real Time Systems, High-Performance Computer Architecture</i>	Progressive Courses	<i>Cyber Physical Systems and Security</i>
------------------------------	--	-----------------------------	--	----------------------------	--

Course Offering Department	Information Technology	Data Book / Codes/Standards	<i>NA</i>
-----------------------------------	-------------------------------	------------------------------------	-----------

Course Objective	This course aims to: – Introduce the principles, architecture, and design methodologies of embedded systems. – Familiarize students with IoT concepts, architectures, and enabling technologies. – Provide hands-on experience in embedded programming using C and hardware description using VHDL/Verilog. – Develop skills in integrating embedded systems with IoT applications through sensors, communication protocols, and cloud platforms. – Equip students with the ability to design, simulate, and test embedded/IoT solutions for real-world applications.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Embedded Systems Definition, characteristics, application domains; Embedded vs general-purpose systems; Examples in industry	3	MO1: Identify and explain various embedded system types and applications
2	Embedded System Architecture Hardware components (Processor, memory, I/O, sensors, actuators), software components (firmware, RTOS); Power considerations	3	MO2: Explain embedded architecture and trade-offs
3	Programming Embedded Systems in C C for embedded systems, device drivers, GPIO control, interrupts, timers, serial communication (UART/SPI/I2C)	6	MO3: Write and test basic embedded C programs for hardware interfacing
4	Digital Design Basics with VHDL/Verilog Combinational and sequential logic; FSMs; Writing and simulating simple designs; Using HDL to model embedded hardware	6	MO4: Design and simulate basic digital modules for embedded applications
5	Introduction to IoT Concepts & Architecture IoT definition, layered architecture, enabling technologies, IoT in various domains	3	MO5: Describe IoT architecture and identify enabling technologies
6	IoT Communication Protocols Physical/MAC layer: Wi-Fi, BLE, Zigbee, LoRa; Application layer: MQTT, CoAP, HTTP; Edge vs cloud computing	3	MO6: Select suitable IoT protocols for given applications
7	Sensors, Actuators & Interfacing Types of sensors and actuators, interfacing with microcontrollers, ADC/DAC, signal conditioning	3	MO7: Interface sensors and actuators with embedded hardware
8	IoT Data Management & Cloud Integration IoT gateways, cloud platforms (AWS IoT, Azure IoT, Google Cloud IoT), data security, privacy considerations	3	MO8: Integrate IoT data with cloud and apply basic security measures
9	Embedded IoT System Design Methodology Requirement analysis, hardware selection, prototyping, testing, deployment	3	MO9: Plan and execute embedded IoT system development
10	Mini Project Work Students design & simulate/implement a small embedded IoT system using C + HDL + IoT integration	6	MO10: Demonstrate an integrated hardware-software IoT prototype
11	Project Presentation & Course Wrap-up Project evaluation, reflections, latest trends in embedded IoT	3	MO11: Present and defend a working embedded IoT solution

	Total	42	
--	-------	----	--

Course Outcome	At the end of the course, students will be able to: CO1: Explain the characteristics, components, and architecture of embedded systems and IoT. CO2: Develop C programs for microcontroller-based embedded applications. CO3: Describe and implement basic digital designs using VHDL/Verilog for embedded hardware components. CO4: Apply IoT protocols, communication standards, and cloud integration for data exchange. CO5: Design, simulate, and demonstrate a small-scale embedded IoT project integrating hardware and software.
-----------------------	--

Learning Resources	Primary Textbooks: 1. Raj Kamal, Embedded Systems: Architecture, Programming and Design, McGraw Hill. 2. Adrian McEwen & Hakim Cassimally, Designing the Internet of Things, Wiley. 3. Frank Vahid & Tony Givargis, Embedded System Design: A Unified Hardware/Software Introduction, Wiley. Supplementary References: 4. Elecia White, Making Embedded Systems: Design Patterns for Great Software, O'Reilly. 5. Perry, VHDL, McGraw Hill. 6. Sam Siewert, Real-Time Embedded Components and Systems with Linux and RTOS, Elsevier. 7. Arshdeep Bahga & Vijay Madisetti, Internet of Things: A Hands-On Approach, Universities Press.
---------------------------	---

Course Code	IT3224N	Course Name	High-Performance Computer Architecture	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic and Circuit Design, Computer Organization and Architecture, Basic programming concepts	Co-requisite Courses	<i>Embedded Systems and IoT, Real Time Systems</i>	Progressive Courses	<i>Cyber Physical Systems and Security</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	By the end of this course, students should be able to: – Understand the design principles and trade-offs in modern high-performance processor architectures. – Analyze the performance implications of advanced instruction-level parallelism (ILP), memory hierarchies, and interconnection networks. – Apply quantitative approaches to evaluate architectural features and system performance. – Examine multicore, manycore, and heterogeneous computing architectures. – Relate architectural advancements to emerging computing domains such as AI/ML, cloud, and HPC.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction & Performance Basics Evolution of high-performance computing, Flynn's taxonomy, quantitative design principles (Amdahl's Law, Gustafson's Law), performance metrics (latency, throughput, CPI), benchmarking.	6	MO1: Explain HPC trends and performance metrics. MO2: Apply Amdahl's and Gustafson's Laws to real scenarios.
2	Instruction-Level Parallelism (ILP) Pipelining review, hazards and solutions, dynamic scheduling (Tomasulo's algorithm), branch prediction, speculative execution, superscalar architectures, VLIW basics.	6	MO3: Analyze ILP techniques and their impact; MO4: Evaluate hazard resolution methods.
3	Memory Hierarchy & Optimization Cache design (mapping, replacement, write policies), multilevel caches, cache coherence, virtual memory, TLB, memory bandwidth, prefetching, non-uniform memory access (NUMA).	6	MO5: Analyze cache performance; MO6: Suggest optimizations for memory hierarchy.
4	Parallelism Beyond ILP Thread-level parallelism (TLP), data-level parallelism (DLP), SIMD/vector processing, GPU architectures, GPGPU programming concepts (CUDA/OpenCL overview).	6	MO7: Compare ILP, TLP, and DLP; MO8: Explain GPU execution models.
5	Multiprocessors & Interconnection Networks Symmetric multiprocessors (SMP), ccNUMA, message passing, interconnection topologies (bus, mesh, torus, hypercube), network routing strategies.	6	MO9: Explain multiprocessor designs; MO10: Analyze interconnection network performance.
6	Advanced Topics in HPC Architecture Heterogeneous computing (CPU+GPU, FPGA accelerators), domain-specific architectures (TPUs, AI accelerators), emerging memory technologies (HBM, NVRAM).	6	MO11: Evaluate heterogeneous architectures; MO12: Relate architectural choices to emerging workloads.
7	Trends & Case Studies Exascale computing, quantum-inspired architectures, ARM vs. x86 in HPC, cloud-scale datacenters; Case studies: NVIDIA Hopper GPU, Apple M-series chips, Fugaku supercomputer.	6	MO13: Critically analyze modern architecture case studies. MO14: Discuss future HPC trends.
	Total	42	

Course Outcome	Upon successful completion, students will be able to: CO1. Explain the principles of pipelining, superscalar, and VLIW architectures. CO2. Analyze cache/memory hierarchy performance and propose optimization strategies. CO3. Evaluate ILP techniques, branch prediction, and speculative execution mechanisms.
-----------------------	---

	CO4. Compare multicore, GPU, and heterogeneous architectures in terms of scalability and performance. CO5. Apply performance metrics (e.g., CPI, throughput, speedup) to assess design trade-offs. CO6. Design small-scale architectural improvements and justify them using simulation or analytical models.
--	--

Learning Resources	Primary Textbooks: 1. John L. Hennessy & David A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann, 2019. 2. David E. Culler & Jaswinder Pal Singh, Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann, 1999. Supplementary References: 3. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, 2nd Edition. 4. Andrew S. Tanenbaum & Todd Austin, Structured Computer Organization, 6th Edition, Pearson. 5. Wen-mei W. Hwu et al., Programming Massively Parallel Processors: A Hands-on Approach, 4th Edition, Morgan Kaufmann, 2022.
---------------------------	---

Course Code	IT3225N	Course Name	Computational Geometry	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Concepts of Computer Graphics, Data Structures and Algorithms</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology		Data Book / Codes/Standards		

Course Objective	Introduce the fundamental concepts, models, and problem-solving paradigms used in computational geometry, with a focus on geometric algorithms and data structures. Develop analytical and design skills for solving geometric problems such as convex hulls, Voronoi diagrams, Delaunay triangulations, and range searching efficiently. Enable students to analyze the complexity and correctness of geometric algorithms, including worst-case, average-case, and output-sensitive analysis.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Computational Geometry: Introduction, degeneracy and robustness, Application domains	3	Explain the scope, significance, and application areas of computational geometry in real-world scenarios. Identify issues of degeneracy and robustness in geometric algorithms and describe methods to address them.
2.	Orthogonal range searching (in brief): kd-tree, range tree, Lower Bounds on Algebraic tree model and Geometric data structures (DCEL)	6	Implement and analyze kd-trees and range trees for orthogonal range queries. Understand and apply the DCEL (Doubly Connected Edge List) for planar subdivisions.
3.	The Maximal Points Problem (closest pair and farthest pair), Geometric searching, Slab method, Range searching	3	Solve maximal points problems including closest and farthest pair computations efficiently. Apply slab decomposition techniques in range searching and analyze their complexity.
4.	Point Location and Triangulation, triangulating monotone polygon	4	Apply point location techniques in planar subdivisions. Perform triangulation of monotone polygons and evaluate its applications in computational geometry.
5.	Convex Hull, Different Paradigms, Voronoi Diagram and Delaunay Triangulation, and Quickhull	6	Construct convex hulls using different algorithmic paradigms including Quickhull. Build Voronoi diagrams and Delaunay triangulations and explain their geometric properties and applications.
6.	Line segment intersection, Linear programming, Intersection of convex polygons, planes	5	Implement algorithms for detecting intersections among line segments. Solve geometric linear programming problems. Determine intersections of convex polygons and 3D planes.
7.	Clustering Point Sets using Quadrees and Applications	3	Implement quadtree-based spatial partitioning for clustering point sets. Analyze performance and apply in relevant computational geometry problems.

8.	Introduction using Basic Visibility Problems, visibility graph and edge and applications to robot path planning	3	Model and solve visibility problems in polygonal environments. Construct visibility graphs and apply them in robot path planning scenarios.
9.	Shape Analysis and Shape Comparison	3	Apply algorithms for comparing and analyzing shapes in 2D geometry. Utilize geometric similarity measures in shape recognition tasks.
10	Intersection and union of rectangles and largest empty space recognition	3	Compute intersections and unions of axis-aligned rectangles. Develop algorithms to find the largest empty space within a given geometric domain.
11.	Some applications and case studies	3	Analyze real-world problems using computational geometry techniques.
	Total	42	

Course Outcome	1. Explain the fundamental concepts, terminologies, and application areas of computational geometry, addressing issues of degeneracy and robustness. Design and implement geometric data structures such as kd-trees, range trees, and DCEL to solve spatial searching problems. 2. Apply algorithmic techniques such as slab decomposition, divide-and-conquer, and sweep-line methods to solve range searching, closest pair, and farthest pair problems. 3. Perform polygon triangulation, point location, and convex hull construction using different algorithmic paradigms, and analyze their time complexity. 4. Construct Voronoi diagrams and Delaunay triangulations, and utilize them in proximity, clustering, and spatial partitioning applications. 5. Apply shape analysis, rectangle intersection/union, and largest empty space detection to solve practical geometry problems. 6. Integrate computational geometry algorithms into real-world case studies, demonstrating problem-solving and optimization skills.
-----------------------	---

Learning Resources	References: 1. Computational Geometry Algorithms and Applications, Authors: de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. 2. Computational Geometry, An Introduction, Authors: Preparata, Franco P., Shamos, Michael 3. Discrete and Computational Geometry, Satyan L. Devadoss & Joseph O'Rourke
---------------------------	---

Course Code	IT3226N	Course Name	Wireless Networks	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Communication Systems, Computer Organization and Architecture	Co-requisite Courses	<i>Embedded Systems and IoT, Artificial Intelligence</i>	Progressive Courses	<i>Cloud Computing and Web Services, Intelligent Transportation and Smart Systems, Cognitive radio networks</i>			
Course Offering Department		<i>Information Technology</i>		Data Book/ Codes/Standards				

Course Objective	• To study about Wireless networks, communication protocols and standards • To study about resource allocation and QoS in wireless networks • To study about different generations of wireless network services, its protocols and applications. • To study about evolution of 4G Networks, its architecture and applications with LTE advanced • To study the importance security in wireless network
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction to wireless networks, history of radio network, cellular architecture	2	Wireless networks and cellular networks introduction
2.	Cellular network architecture and standards	4	Cellular communication standards
3.	Telecommunication union spectrum allocation, mobile radio propagation, satellite communication	4	ITU standards for wireless networks, radio wave communication
4.	Wireless communication technologies: Digital modulation, spread spectrum communication	4	Data transmission and communication techniques
5.	Multiple access scheme for wireless communication, generation of mobile communications and standards, networking and design	6	Channel allocation and topology generation in wireless networks
6.	Wireless networking and protocol design: PSTN, GSM, WLAN, WLL, WAP, WWW	8	Networking and protocol standards for various wireless networks

7.	Modern wireless networks: cellular networks, 3GPP standards, LTE advanced	6	Future generation wireless networking
8.	Wireless network security	4	Security and authentication in wireless network
9.	Wireless application environment, introduction to J2ME, application specific GUI development	4	Design of application software
	Total	42	

Course Outcome	• Design and implement wireless network environment for any application using latest wireless protocols and standards. • Implement different types of applications for smart phones and mobile devices with latest network strategies • Compare and contrast multiple division techniques, mobile communication systems, and existing wireless networks. • Classify network protocols, ad hoc and sensor networks, wireless MANs, LANs and PANs
-----------------------	--

Learning Resources	1. Wireless and Mobile Networks, Concepts and Protocols, 2ed Kindle Edition, by Sunilkumar S. Manvi, Mahabaleshwar S. Kakkasageri 2. Wireless Networks (Computer Science, Technology and Applications), by Dr S Anandamurugan, P S Nandhini, Nova Science Publishers Inc 3. MOBILE AND WIRELESS COMMUNICATIONS: AN INTRODUCTION, by Gordon A. Gow and Richard K. Smith, Open University Press
---------------------------	---

Course Code	IT3227N	Course Name	Computer Vision	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Data Structures and Algorithms, Probability and Statistics, Linear Algebra and Matrix Theory, Programming in Python / C++, Basic Image Processing</i>	Co-requisite Courses	<i>Machine Learning, Numerical Methods</i>	Progressive Courses	
Course Offering Department	Information Technology			Data Book / Codes/Standards	

Course Objective	1. Introduce the mathematical foundations, algorithms, and techniques for analyzing and interpreting visual data. 2. Develop the ability to extract, represent, and match features from images and videos for object recognition and scene understanding. 3. Enable students to design computer vision pipelines for real-world applications such as face recognition, motion tracking, and 3D reconstruction. 4. Equip students with skills to apply deep learning approaches for vision problems and integrate them with classical methods.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction to Computer Vision ● Overview of computer vision and its applications (autonomous vehicles, surveillance, AR/VR, medical imaging) ● Vision vs. image processing ● Human vision system inspiration ● Image formation: camera models (pinhole, perspective projection, lens distortion) ● Radiometry basics, image sensing, and digitization	4	Explain the fundamentals of image formation, representation, and transformations. Describe applications of computer vision in robotics, medical imaging, surveillance, and autonomous systems.
2.	Image Filtering and Enhancement ● Convolution, correlation, and separable filters ● Smoothing and sharpening filters ● Edge detection: Sobel, Prewitt, Laplacian, Canny ● Image pyramids and scale-space representation ● Histogram equalization and contrast stretching	5	Apply convolution and correlation operations for image filtering. Implement smoothing and sharpening filters to enhance image quality. Compare and evaluate edge detection operators (Sobel, Prewitt, Laplacian, Canny).

			Apply histogram equalization and contrast stretching for image enhancement.
3.	Feature Detection and Description • Corners and interest points: Harris, Shi-Tomasi • Blob detection: LoG, DoG • Feature descriptors: SIFT, SURF, ORB • Feature matching: distance metrics, ratio test, RANSAC • Invariance to scale, rotation, and illumination	6	Detect corners, interest points, and blobs using established operators (Harris, Shi-Tomasi, LoG, DoG). Construct robust feature descriptors (SIFT, SURF, ORB) for object recognition. Perform feature matching using distance metrics, ratio tests, and RANSAC. Assess feature invariance to scale, rotation, and illumination.
4.	Geometric Vision and Camera Models • Epipolar geometry, fundamental and essential matrices • Stereo vision, depth estimation • Homography and planar transformations • Camera calibration (Zhang's method) • 3D reconstruction from multiple views	6	Explain the concepts of epipolar geometry and compute fundamental and essential matrices. Apply stereo vision techniques for depth estimation.

			Use homography and planar transformations for perspective correction. Implement camera calibration (Zhang's method) to estimate intrinsic and extrinsic parameters. Reconstruct 3D structure from multiple views.
5.	Motion Analysis and Tracking • Optical flow (Horn–Schunck, Lucas–Kanade) • Background subtraction and modeling (Mixture of Gaussians) • Object tracking: Mean-shift, CamShift, Kalman filter, Particle filter • Action recognition basics	5	Estimate motion fields using optical flow techniques (Horn–Schunck, Lucas–Kanade). Apply background subtraction and Gaussian mixture models for foreground detection. Implement tracking algorithms such as Mean-shift, CamShift, Kalman filter, and Particle filter. Demonstrate the basics of action recognition from motion sequences.
6.	Object Detection and Recognition • Sliding window detectors, HOG features	6	Implement classical object detection

	• Viola–Jones face detection • Introduction to convolutional neural networks (CNN) for object detection • Modern detectors: R-CNN, Fast R-CNN, YOLO (overview) • Transfer learning for vision tasks		methods (sliding windows, HOG features). Apply Viola–Jones algorithm for face detection. Explain the role of convolutional neural networks (CNNs) in object detection. Compare modern detectors such as R-CNN, Fast R-CNN, and YOLO in terms of accuracy and speed. Apply transfer learning techniques to vision recognition tasks.
7.	Advanced Topics and Applications • Image segmentation: Thresholding, region growing, graph cuts, watershed, semantic segmentation • Shape analysis: contours, Hough transform for line/circle detection • Introduction to 3D vision: point clouds, surface fitting • Case studies: medical imaging, autonomous navigation, AR/VR • Ethical and societal issues in computer vision	6	Implement image segmentation methods (thresholding, region growing, graph cuts, watershed). Perform shape analysis using contours and Hough transform for line and circle detection. Explore 3D vision techniques using point

			clouds and surface fitting. Analyze case studies in medical imaging, AR/VR, and autonomous navigation. Critically evaluate ethical and societal issues in deploying computer vision systems.
	Total	42	

Course Outcome	1: Explain fundamental concepts, mathematical models, and computational techniques in computer vision, including image formation, transformations, and feature extraction. 2: Apply image preprocessing, filtering, and enhancement methods to improve the quality and interpretability of images for vision applications. 3: Implement and evaluate classical computer vision algorithms for object detection, recognition, and tracking. 4: Utilize modern machine learning and deep learning approaches, such as CNN-based architectures, to solve complex vision problems. 5: Analyze and select appropriate computer vision techniques for applications in areas such as robotics, medical imaging, autonomous systems, and surveillance.
-----------------------	--

Learning Resources	Books 1. Richard Szeliski, <i>Computer Vision: Algorithms and Applications</i>, Springer. 2. E.R. Davies, <i>Computer & Machine Vision</i>, Academic Press. 3. Gary Bradski & Adrian Kaehler, <i>Learning OpenCV</i>, O'Reilly. Online • OpenCV documentation (https://docs.opencv.org) • Stanford CS231n lecture notes • PyImageSearch tutorials Software Tools • Python with OpenCV, NumPy, Matplotlib, Scikit-image • MATLAB Image Processing Toolbox
---------------------------	---

Course Code	IT3228N	Course Name	Natural Language Processing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Machine Learning</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	• Understanding and fundamental concepts of Natural language processing • Understand text classification and machine translation • Learns modern deep learning approaches for NLP • Will understand real world application of deep learning
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction, Regular Languages and Finite state automata, Finite state transducers	5	Understand basics of NLP and FSM
2	Classical text categorisation: Naive Bayes, Logistic Regression	4	Learns classification of languages
3	Sentiment mining and Lexical Learning	4	Understands sentiment mining
4	Vector spaces in information Retrieval	4	Learns Vectors spaces and information retrieval techniques
5	Deep learning for NLP	6	Learn various deep learning models for NLP
6	LLM for NLP, retrieval augmentation in LLM, Efficient inference for LLM, Reasoning with LLMs	6	Learns various LLM models and its uses for NLP
7	Representation Discovery: Word2Vec, Glove N-Gram features with CNN, RNN for variable length sequence, Neural CRF for sequence learning, N-Gram language model, Neural language models	8	Understands representation of languages and its uses

8	Statistical Natural Language parsing	5	Learns statistical natural language parsing and its applications
	Total	42	

Course Outcome	- Proficiency in statistical machine learning model and deep learning model - Understand text processing techniques - Evaluating NLP system performance on various tasks
-----------------------	--

Learning Resources	- Foundations of Statistical Natural Language Processing by Christopher Manning - Speech and Language Processing by Jurafsky and Martin - Practical Natural Language Processing by Vajjala et al - Neural Network Methods for Natural Language Processing, Yoav Goldberg, Morgan and Claypool..
---------------------------	--

Course Code	IT3261N	Course Name	Internet of Things	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Concepts of Embedded Systems</i> <i>Basics of Wireless and Computer Networks</i> <i>Operating Systems Fundamentals</i>	Co-requisite Courses	<i>Microcontrollers and Embedded Systems</i> <i>Network Security</i>	Progressive Courses	<i>Cyber-Physical Systems</i> <i>IoT Security and Privacy</i> <i>Smart Systems and Edge Computing</i> <i>Cloud Computing and Big Data for IoT</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	NA

Course Objective	To introduce the fundamental concepts, architectures, and technologies of the Internet of Things (IoT). This course enables students to understand embedded systems integration, networking, data analytics, and security considerations in IoT systems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to the internet of things	4	Understand the scope, applications, and evolution of IoT.
2	Basic of Internet technologies	6	Gain knowledge of internet protocols and their role in IoT systems.
3	IoT standards, open-source vs closed source	6	Evaluate and compare existing IoT platforms and standards.
4	Introduction of embedded computing	5	Understand the role of embedded systems in IoT architecture.
5	IP as the IoT network layer	5	Analyze how IP-based communication enables IoT connectivity.
6	Getting started with API	4	Learn to use web APIs for data exchange and device communication.
7	Data and analytics for IoT	6	Understand data flow in IoT and apply basic analytics methods.
8	IoT Data security	6	Identify key security threats and implement basic protections in IoT systems.
	Total	42	

Course Outcome	Understand the fundamentals of object-oriented design principles and their role in software development. Apply object-oriented modelling using UML for different views of system architecture. Design and implement object-oriented systems using C++/Java. Analyse system behaviour using structural, behavioural, and dynamic models. Utilize design patterns and best practices for software reusability and maintainability. Evaluate and ensure system usability and user satisfaction. Gain exposure to object-oriented databases and distributed object systems.
-----------------------	--

Learning Resources	Books: 1. Designing the Internet of Things By Adrian McEwen, Hakim Cassimally, J Wiley 2. Precision - Principles, Practices, and Solutions for the Internet of Things, Book by Timothy C K Chou, McGraw-Hill 3. Getting Started with the Internet of Things, Cuno Pfister, O Reilly media 4. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, Cisco Press
---------------------------	---

Course Code	IT3271N	Course Name	Computer Networks Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Computer programming, Operating system,</i>	Co-requisite Courses		Progressive Courses	<i>Information System and Security</i>
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	1. Provides Hands on experience with Networking concepts 2. Provide understanding to implement fundamental network principles, protocols 3. Helps to understand how to write client server applications using basic protocols
-------------------------	---

Module	Module Name and Topics	Duration (class-hour)	Module Outcome
1	Setting up of LAN	3	Understands different cables and connectors to set up a LAN
2	Internetworking using a router	6	Understands network devices and their configurations
3	Understanding and implementation of basic socket programming	9	Learns socket programming
4	Network analysis using raw sockets, Use of different Tools: Wireshark, iperf3	6	Learns network analysis with tools and using programming
5	Writing client server programs under various scenarios	12	Write client -server applications
6	Study and analysis of congestion control in a real scenario	3	Learns congestion control aspects
7	Implementation of denial of service and other attacks	3	Understands DoS attacks and its implications
	Total	42	

Course Outcome	1. Students learn to physical set up a local LAN by understanding network cables and connectors (CAT5, RJ45) 2. Practical knowledge in configuring and differentiating network devices such as hubs, switches, routers etc. 3. Understands to learn client-server applications programming (socket programming) using protocols like TCP and UDP to transfer data between two hosts 4. Students familiar with tools like wireshark to analyze network traffic
-----------------------	--

Learning Resources	1. Unix Network Programming by Richard Stevens 2. Hands-On Network Programming with C: Learn socket programming in C and write secure and optimized network code by Lewis Van Winkle 3. Wireshark User's Guide, (https://docs.google.com/document/d/1SL17FVslb1v_JuI26S1T0X3bp_3J9DG4/edit)
---------------------------	---

Course Code	IT3272N	Course Name	Machine Learning Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	<i>Python Programming experience</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To provide hands-on experience in implementing, experimenting with, and evaluating machine learning algorithms across supervised, unsupervised, and reinforcement learning paradigms. The course aims to strengthen students' ability to preprocess data, select suitable models, fine-tune hyperparameters, and assess performance using appropriate metrics.
-------------------------	--

Module	Module Name and Topics	Duration (class-hour)	Module Outcome
1	Basics of neural networks and related problems	5	Implement perceptrons and multilayer neural networks using suitable libraries. Apply activation functions and understand their role in learning.

2	Solving regression problems	3	Preprocess datasets for regression tasks and handle missing or noisy data. Evaluate regression models using appropriate performance metrics (e.g., RMSE, MAE).
3	Supervised learning for classification	6	Implement classification algorithms such as decision trees, SVM, and k-NN. Train and validate models using cross-validation techniques. Apply confusion matrices and classification metrics (precision, recall, F1-score) for evaluation. Compare the performance of multiple classification algorithms on the same dataset.
4	Feature selection techniques and applications	6	Apply filter, wrapper, and embedded methods for feature selection. Evaluate the effect of feature selection on model accuracy and training time.
5	Unsupervised learning - clustering	6	Implement K-means, hierarchical, and density-based clustering algorithms. Evaluate clustering quality using internal and external metrics. Visualize clusters and interpret results in the context of application domains.
6	Ensemble techniques - Bagging and Boosting	6	Implement ensemble learning techniques such as Random Forest (bagging) and AdaBoost/Gradient Boosting. Analyze the effect of ensemble size and base learner

			complexity on performance. Compare ensemble methods with single learners for given datasets.
7	Bayesian network	6	Construct Bayesian networks for probabilistic reasoning. Apply conditional probability concepts for inference in Bayesian networks.
8	Reinforcement learning	4	Implement basic reinforcement learning algorithms such as Q-learning. Define environments, states, actions, and rewards for given problems.
	Total	42	

Course Outcome	1. Implement and evaluate neural networks, regression models, and classification algorithms for real-world datasets. <i>(Applying & Analyzing)</i> 2. Apply feature selection and dimensionality reduction techniques to improve model efficiency and interpretability. <i>(Applying & Evaluating)</i> 3. Implement unsupervised learning methods such as clustering and analyze their results using suitable evaluation metrics. <i>(Applying & Analyzing)</i> 4. Apply ensemble techniques like bagging and boosting to enhance predictive performance. <i>(Applying & Evaluating)</i> 5. Construct Bayesian networks for probabilistic reasoning and perform inference on real-world problems. <i>(Applying & Evaluating)</i> 6. Implement basic reinforcement learning algorithms and evaluate agent performance in defined environments. <i>(Applying & Analyzing)</i>
-----------------------	--

Learning Resources	Python (Anaconda Distribution) – for integrated ML libraries and data science tools. Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn – for ML, data handling, and visualization. Keras, TensorFlow, PyTorch – for neural networks and deep learning. Jupyter Notebook – for interactive coding and documentation. Google Colab – for cloud-based execution with GPU/TPU support. Scikit-learn Documentation: https://scikit-learn.org <i>(Python library for ML algorithms with examples)</i> Keras Documentation: https://keras.io <i>(Neural network library with tutorials)</i>
---------------------------	---

	TensorFlow Tutorials: https://www.tensorflow.org/tutorials (<i>Hands-on examples for deep learning and RL</i>) PyTorch Tutorials: https://pytorch.org/tutorials/ (<i>Deep learning framework tutorials</i>) Stanford CS229 Machine Learning: https://cs229.stanford.edu (<i>Lecture notes and assignments</i>) Trevor Hastie, Robert Tibshirani, Jerome Friedman – <i>The Elements of Statistical Learning</i>, Springer, 2017. (<i>Comprehensive coverage of ensemble learning, regression, and classification</i>) Kevin P. Murphy – <i>Machine Learning: A Probabilistic Perspective</i>, MIT Press, 2012. (<i>Bayesian networks, probabilistic inference</i>) Aurélien Géron – <i>Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow</i>, O'Reilly, 2022. (<i>Practical, code-focused ML implementation</i>) introduce the fundamental concepts, models, and problem-solving paradigm
--	--

Course Code	IT4101N	Course Name	Information and Systems Security	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	<i>Number Theory</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	To learn present-day security goals and practices
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Basics of Security and Cryptography: (3L) Three major goals of security, major security attacks, security services related to three goals of security, security mechanisms, cryptography and steganography. Cryptanalysis attacks, classes of cryptanalysis attacks.	4	Goals of Security
2	Mathematical Background: (3L) Introduction to Number theory, Modular arithmetic, Prime number generation, Primality Testing, GCD, Euclidean Algorithm, Extended Euclidean Algorithm, Chinese Remainder Theorem, Fermat's Little Theorem and Euler's Theorem, Index of Coincidence.	4	Number Theoretic Basis
3	Private Key Cryptography: Symmetric Key Encryption. Definitions. Block ciphers and Stream ciphers. Substitution Ciphers and Transposition Ciphers. Traditional Ciphers vs. Modern Day Ciphers. Chosen-Plaintext Attack. Chosen-Ciphertext Attack, Known-Plaintext Attack. Known-Ciphertext Attack, Pattern Analysis Attacks, Statistical Attacks. Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES. Modes of operation of block ciphers (ECB, CBC, CFB, OFB, Counter modes).	8	Symmetric Key Cryptosystems
4	Public Key Cryptosystems: RSA, ElGamal, Elliptic curve cryptosystems, Public Key Infrastructure (PKI), Digital Signatures, Digital Certificates, Key Management and Key Distribution techniques.	8	Public Key Cryptosystems
5	Message Authentication and Integrity Verification: Message Authentication Codes, Modification Detection Codes, Hash Functions, Cryptographic Hash Functions, Merkle Damgard Scheme, Preimage Resistance, Second Preimage Resistance,	10	Message Authentication and Hash

	Collision Resistance, Random Oracle Model. Digital Signatures, RSA Digital Signature.		
6	Entity Authentication: Differentiate between message authentication and entity authentication, Data origin vs. entity authentication, Verification categories, Password based authentication, Challenge-response based authentication, Zero-knowledge authentication, biometric authentication.	4	Entity Authentication
7	Key Management: Key Distribution Center (KDC), Symmetric key agreement protocol, Kerberos as a KDC, Certification Authorities for Public Keys, Role of Public Key Infrastructure.	4	Key Management
	Total	42	

Course Outcome	- To learn present-day private- and public key cryptosystems - Message authentication and hash - Key management and entity authentication
-----------------------	---

Learning Resources	Reading: 1. Behrouz A. Forouzan and D. Mukhopadhyay - Cryptography & Network Security, McGraw Hill. 2. Douglas R. Stinson, Cryptography: Theory and Practice, Chapman and Hall. 3. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall.
---------------------------	---

Course Code	IT4121N	Course Name	Distributed Algorithms	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Algorithms</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To understand the basics of distributed algorithms
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Characterization of Distributed Systems, Network OS vs. distributed OS, Middleware	6	Stepping into distributed systems
2	Remote Procedure Call: Design issues for RPCs, Case study: Sun RPC; Remote method invocation and JAVA	4	Understanding of distributed systems
3	Logical clock; Election algorithms; Lower bounds for synchronous networks; Synchronization; Mutual exclusion algorithms	8	Understanding basics of distributed algorithms
4	Distributed Shared Memory: Shared memory, Consistency models, Design issues, Case studies: Ivy/Munin/Treadmarks	6	Understanding of consistency
5	Distributed Consensus and Fault tolerance	6	Understanding fault tolerance
6	Distributed Naming Services: Names, addresses, routes, capabilities, Naming facilities, name distribution, name resolution, Migration	4	Understanding role of names in distributed systems
7	Security in Distributed Systems: Basic concepts of Cryptographic techniques	4	Understanding security in distributed systems
8	Distributed Algorithms for Mobile Environment	4	
	Total	42	

Course Outcome	Understanding of basics of distributed algorithms
-----------------------	---

Learning Resources	References: 1. Nancy Lynch. Distributed Algorithms. Elsevier. 2. Gerard Tel. Introduction to Distributed Algorithms. Cambridge University Press. 3. Ajay D. Kshemkalyani and Mukesh Singhal. Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press 4. Sukumar Ghosh. Distributed Systems: An Algorithmic Approach . CRC Press, Second Edition
---------------------------	---

Course Code	IT4122N	Course Name	Pattern Recognition	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Signals, and Systems, Image Processing, Discrete Math</i>	Co-requisite Courses		Progressive Courses	<i>Image Analysis, Computer Vision</i>
Course Offering Department	Information Technology			Data Book / Codes/Standards	

Course Objective	To understand the concept and fundamentals of Pattern Recognition Math. prelim for PR problem. Parametric and non parametric approaches used in PR problems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Pattern Recognition: Introduction and overview of different approaches to PR, decision boundaries, discriminant functions	2	Understanding of pattern recognition and connection with AI and Image processing and analysis
2.	Mathematical preliminaries for PR Probability: independence of events, conditional and joint probability, Bayes theorem Random Processes: Stationary and non-stationary processes, Expectation, Autocorrelation, Cross-Correlation, spectra. Linear Algebra: Inner product, outer product, inverses, eigenvalues, eigen vectors, singular values, singular vectors.	6	Linear algebra, vector-matrix theory, signal/vector representation, dimensionality reduction
3.	Bayes Decision Theory: Minimum-error-rate classification. Classifiers, Discriminant functions, Decision surfaces. Normal density and discriminant functions. Discrete features	6	Statistical analysis for pattern classifier
4.	Parameter Estimation Methods : Maximum-Likelihood estimation :Gaussian case. Maximum a Posteriori estimation. Bayesian estimation: Gaussian case.	4	Detection and estimation theory , role in pattern classification
5	Unsupervised learning and clustering - Criterion functions for clustering. Algorithms for clustering: K-Means, Hierarchical and other methods. Cluster validation. Gaussian mixture models, Expectation-Maximization method for parameter estimation. Maximum entropy estimation. Sequential Pattern Recognition. Hidden Markov	8	Pattern recognition based features grouping, clustering and information theory, ML and RL methods

	Models (HMMs). Discrete HMMs. Continuous HMMs		
6.	Nonparametric techniques for density estimation. Parzen-window method. K-Nearest Neighbour method.	4	Non-parametric methods for pattern recognition, recent ML based methods
7	Dimensionality reduction: Principal component analysis - its relationship to eigen analysis. Fisher discriminant analysis - Generalised eigen analysis. Eigenvectors/Singular vectors as dictionaries.	4	PCA. SVD on dimensionality reduction and role in PR
8.	Artificial neural networks: Multilayer perceptron - feedforward neural network. A brief introduction to deep neural networks, convolutional neural networks, recurrent neural networks.	8	Introduction to ANN and DL (ANN, CNN and RNN), classifier design and usage in PR
	Total	42	

Course Outcome	CO1: Intelligent and automated scheme of Pattern Recognition CO2: Mathematical Preliminaries on Pattern Recognition CO2: Use of supervised, unsupervised methods, statistical and information theoretic approaches for design and analysis of Pattern Recognition CO3: Dimensionality reduction and its role on Pattern Recognition CO4: Introduction to ANN, CNN, RNN and their roles in Pattern Recognition
-----------------------	---

Learning Resources	Books: 1 R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 20012. 2. S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009 3. J. T.Tou and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, London 4. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006
---------------------------	---

Course Code	IT4123N	Course Name	Cloud Computing and Web Services	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Operating Systems, Computer Networks</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	To understand the following: 1. Basics of cloud computing 2. Advantages and disadvantages of clouds 3. Cloud backup and solutions 4. Different cloud-computing services 5. Concepts of virtualization
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	INTRODUCTION Definition, private, public and hybrid cloud. IaaS, PaaS, SaaS. Benefits and challenges of cloud computing.	4	Get introduced to cloud computing
2	CLOUD STORAGE INFRASTRUCTURES Storage strategy. Securing storage in cloud environments. Monitoring, management, security regulation. Storage networks.	8	Understand different cloud storage strategies
3	CLOUD SECURITY Introduction to cloud security. Infrastructure security: Network and host level security, application level security. Data security: Privacy and security issues, jurisdictional issues involving data location, audits. Identity and access management: Access policies and control, trust, reputation, risk, authentication, authorization.	10	Identify security issues and how to resolve them
4	IaaS Definition, virtualization and its different approaches, virtual machine and image, resource virtualization, storage as a service, hypervisors, software-defined networks. PaaS Basics, SOA, SOAP, REST, Microservices, cloud platform and management. SaaS Introduction. Cloud-based applications.	12	Know about infrastructures, platforms and software provided as a service by cloud
5	Case studies and cloud solutions including cloud data platforms (e.g., Databricks, Snowflake)	8	Case studies and cloud solutions
	Total	42	

Course Outcome	On successful completion of this course, students will learn 1. Basics of cloud computing 2. Advantages and disadvantages of clouds 3. Cloud backup and solutions 4. Different cloud-computing services 5. Concepts of virtualization
-----------------------	--

Learning Resources	Books: 1. Cloud Computing: A Practical Approach. Velte, Toby, Anthony Velte, and Robert Elsenpeter. McGraw-Hill, Inc., 2009. 2. Cloud Computing: Principles, Systems and Applications. Editors: Nikos Antonopoulos and Lee Gillam, Springer, 2012. 3. Cloud Computing: Principles and Paradigms. Editors: Raj Kumar Buyya, James Broberg and Andrzej Goscinski. Wiley Press, 2011. 4. Cloud Application Architectures. George Reese. O'Reilly Media, Inc., 2009. 5. Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010.
---------------------------	---

Course Code	IT4124N	Course Name	Parallel and Distributed Systems	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Computer Organization & Architecture, Operating Systems, Computer Networks</i>	Co-requisite Courses	<i>Distributed Algorithms</i>	Progressive Courses	<i>Projects</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	1. To understand high-performance computing 2. To get familiar with different distributed computing techniques
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Introduction to High Performance Computing: Distributed System models, Grid Computing, Cloud Computing, Cluster Computing, Super Computing, etc.	4	Getting familiar with high

			performance computing architecture
2.	Computing Architectures, Flynn's and Handler's classifications, Multicore Processors, Scalable Multiprocessors	6	Understanding advanced computing architectures
3.	Distributed Systems and Middleware, Introduction to Sun RPC and JAVA RMI	4	Understanding of distributed computing
4.	Distributed storage and File systems.	4	Understanding of distributed computing
5.	Concurrency and Consistency models, Shared memory and Distributed memory.	6	Understanding of distributed computing
6.	Computing Clouds, Service-Oriented Architecture	6	Understanding of distributed computing
7	Grid Computing Systems, Peer-to-peer computing and Overlay graphs	2	Understanding of distributed computing
8	Load balancing.	4	Understanding of distributed computing
9	Fault tolerance.	4	Understanding of fault tolerance
10	Security and Privacy	2	Understanding of security in distributed systems
	Total	42	

Course Outcome	Understanding of high-performance computing Getting familiar with different distributed computing techniques
-----------------------	---

Learning Resources	1. Kai Hwang, Jack Dongarra and Geoffrey C. Fox. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet. 2. Andrew S. Tannenbaum and Maarten van Steen. Distributed Systems: Principles and Paradigms. Prentice Hall, Second Edition. 3. M. J. Quinn. Parallel Computing: Theory and Practice, McGraw Hill. 4. Sukumar Ghosh. Distributed Systems: An Algorithmic Approach . CRC Press, Second Edition
---------------------------	--

Course Code	IT4125N	Course Name	Bioinformatics	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses		Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	The objective of the course is to give an insight into Basics of bioinformatics and application of various computational methods to deal with biology, medical problems.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to bioinformatics, biological sequence/structure, Central dogma of Molecular Biology, Genome Projects, Pattern recognition and prediction, Folding problem, Sequence Analysis, Homology and analogy. 5 All basics of the subjects will be known and also learn the scope of the bioinformatics course.	6	All basics of the subjects will be known and also learn the scope of the bioinformatics course.
2	Classical algorithms in pattern matching and bioinformatics, exact matching problem, suffix trees, dynamic programming	6	Able to know all the classical approaches in various methods of bioinformatics
3	Local alignment gapped and ungapped global alignment. Motif finding: motif models, finding occurrence of known sites, discovering new sites. Gene Finding: predicting reading frames, maximal dependence decomposition.	12	Understand Alignment methods and its applications Able to understand Motif finding in the sequence and its applications
4	Analysis of DNA microarray data using hierarchical clustering, model-based clustering, expectation-maximization clustering, Bayesian model selection.	9	Able to understand Microarray data preparation and processing for its analysis

	Application of Computational techniques on gene expression data, EST searches. Case studies	9	Able to understand the Solving various real time or research papers related in bioinformatics Expressed sequenced tags will be able to understand
	Total	42	

Course Outcome	Upon successful completion of this course, students will: • Have a broad understanding of Basics and understanding of various tasks of bioinformatics that uses computational methods. • Helps us with interdisciplinary work and in biology with the help of computational tools and techniques, approaches.
-----------------------	---

Learning Resources	Books: 1. Arthur M. Lesk, Introduction to bioinformatics, OUP, 2014 2. Mount, David W., and David W. Mount. <i>Bioinformatics: sequence and genome analysis</i> . Cold Spring Harbor, NY: Cold spring harbor laboratory press, 2001. 3. Gregory R. Grant, Warren J. Ewens, Statistical Methods in Bioinformatics: An Introduction, Springer
---------------------------	---

Course Code	IT4126N	Course Name	Internet Technology	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Computer Networks</i>	Co-requisite Courses	<i>Operating Systems</i> <i>Software Engineering</i>	Progressive Courses	<i>Cloud Computing</i> <i>Web Security and Privacy</i> <i>Internet of Things</i> <i>Stack Development</i>
Course Offering Department	<i>Information Technology</i>		Data Book /	NA	

		Codes/Standards	
--	--	------------------------	--

Course Objective	To provide a comprehensive understanding of the structure, protocols, and tools used in the modern Internet and web-based systems, and to enable students to design, develop, and deploy web-based distributed applications using contemporary technologies.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Review of TCP/IP Protocol Stack: IPv4, IPv6, TCP, UDP, ARP, ICMP, SMTP etc.	4	Understand the internet protocol stack and its core communication mechanisms.
2.	Internet as a Distributed System, Transparency and Openness in Internet, RFCs	2	Grasp the design principles of internet as a large-scale distributed system.
3.	Naming in Internet, Name servers, ISP, Governing bodies	2	Learn how the internet's naming system and authorities function.
4.	World Wide Web as a Distributed Document Based System, Client Server Architecture in Web, Browser	2	Understand WWW structure and the browser's role in accessing content.
5.	Web Technologies: Three Tier Web Based Architecture; Jsp, Asp, J2ee, .Net Systems, TypeScript	4	Explore server-side technologies used in web application development.
6.	Document Model: Markup languages, Document types, Designing Website, HTML, Sgml, DTD, DHTML	4	Learn markup languages and best practices for website design.
7.	Cascading Style Sheets: Syntax ,Class Selector, Id Selector Dom (Document Object Model) & Dso (Data Source Object) Approaches To Dynamic Pages: Cgi, Java Applets, Plug Ins, Active X, Java Script – Java Script Object Model, Variables-Constant – Expressions, Conditions-Relational Operators- Data Types – Flow Control – Functions & Objects-events and event handlers – Data type Conversion & Equality – Accessing HTML form elements Basic of XML	4	Create dynamic and interactive content using CSS and JavaScript.

8.	Communication: HTTP, Web Clients and Servers, Proxies, port, URL	6	Examine web communication mechanisms including HTTP and proxies.
9.	JAVA – a language of Internet, Client-side and Server-side Programming	4	Understand Java's role in building web-based applications.
10.	JAVA – a language of Internet, Client-side and Server-side Programming	4	Develop and deploy Java-based web applications.
11	Website Planning & Hosting, Introduction to Internet of Things and web	4	Plan and host websites; learn how IoT integrates with web systems.
12	Recent trends in Web Technology: OAuth 2.0, OpenID Connect (OIDC)	2	Keep pace with emerging web technology trends and innovations.
	Total	42	

Course Outcome	Describe core Internet protocols and their operation within the TCP/IP stack. Explain the distributed nature of the Internet and the principles guiding its evolution. Develop and design web-based systems using mark-up languages, client-side and server-side scripting. Implement and integrate web applications using Java and modern frameworks. Analyse web communication and hosting mechanisms for deploying scalable web applications. Evaluate current web technologies and trends including IoT and dynamic web content.
-----------------------	--

Learning Resources	References: 1. Jeffrey C. Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education 2. Andrew S. Tannenbaum and Maarten van Steen. Distributed Systems: Principles and Paradigms. Prentice Hall, 2nd Edition 3. Bruce Croft, Donal Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Pearson Education 4. Godbole and Kahate, Web Technology, TMH 5. Internet & Intranet Engineering, - Daniel Minoli, TMH.
---------------------------	---

Course Code	IT4127N	Course Name	Deep Learning	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Computer Networks</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	This course aims to provide a comprehensive understanding of neural networks, from basic perceptrons to deep and recurrent architectures. It focuses on training strategies, optimization methods, and regularization techniques to improve model performance. Students will also explore advanced architectures such as convolutional, generative, and multi-view deep learning models, preparing them to design and implement state-of-the-art neural network solutions.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Basics: Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability. Convergence theorem for Perceptron Learning Algorithm.	4	fundamental concepts of neurons, perceptrons, and learning algorithm
2.	Feedforward Networks: Multilayer Perceptron, Gradient Descent, Backpropagation, Empirical Risk Minimization, regularization, autoencoders.	3	Apply multilayer perceptrons, backpropagation, and regularization in feedforward architectures.
3.	Deep Neural Networks: Difficulty of training deep neural networks, Greedy layerwise training.	3	Recognize challenges in training deep networks and employ greedy layer-wise training.
4.	Better Training of Neural Networks: Newer optimization methods for neural networks (Adagrad, adadelata, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).	6	Utilize advanced optimization and regularization techniques to improve neural network performance.

5.	Recurrent Neural Networks: Back propagation through time, Long Short Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs	4	Implement recurrent architectures including LSTM, GRU, and bidirectional RNNs.
6.	Convolutional Neural Networks: LeNet, AlexNet	4	Apply CNN architectures like LeNet and AlexNet for image and pattern recognition tasks.
7.	Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.	6	Design and train generative models such as RBMs and deep Boltzmann machines.
8.	Recent trends: Variational Autoencoders, Generative Adversarial Networks, Multi-task Deep Learning, Multi-view Deep Learning	6	Explore and apply cutting-edge approaches like autoencoders, GANs, and multi-task deep learning.
9.	Applications: Vision, NLP, Speech	6	
	Total	42	

Course Outcome	After going through the course students will be able to 1. Understand the fundamentals of neural networks, perceptrons, learning algorithms, and training principles. 2. Apply feedforward, deep, recurrent, and convolutional architectures to solve real-world problems. 3. Evaluate different optimization, regularization, and training strategies for improving neural network performance. 4. Design and implement advanced models such as generative, autoencoder, and multi-task deep learning systems. 5. Analyze recent trends in deep learning to develop innovative solutions using state-of-the-art architectures.
-----------------------	--

Learning Resources	References: 1. Jeffrey C.Jackson, "Web Technologies--A Computer Science Perspective", Pearson Education 2. Andrew S. Tannenbaum and Maarten van Steen. Distributed Systems: Principles and Paradigms. Prentice Hall, 2nd Edition 3. Bruce Croft, Donal Metzler and Trevor Strohman. Search Engines: Information Retrieval in Practice. Pearson Education 4. Godbole and Kahate, Web Technology, TMH 5. Internet & Intranet Engineering,- Daniel Minoli, TMH.
---------------------------	--

Course Code	IT4128N	Course Name	Data Sciences	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Basic programming (preferably in Python) Mathematics and statistics	Co-requisite Courses	Introduction to Programming Probability and Statistics	Progressive Courses	Machine Learning Deep Learning Artificial Intelligence Data Mining Big Data Analytics
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	NA

Course Objective	To introduce students to the fundamental concepts, tools, and practices of data science, enabling them to perform data analysis, build simple machine learning models, and responsibly communicate insights drawn from data using Python
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Data Science	2	Understand the data science lifecycle and real-world applications.
2	Setting Up the Environment (Python, Anaconda, Colab)	2	Set up a practical coding environment for data science.
3	Python Basics: Data types, loops, control structures	3	Write basic Python programs relevant for data tasks.
4	Data Handling with Pandas	4	Manipulate tabular data using Pandas.
5	Data Cleaning	3	Handle missing values, type conversions, and data inconsistencies.
6	Data Exploration and Visualization: ggplot, Matplotlib, Seaborn	4	Perform EDA and visualize data using Matplotlib and Seaborn.
7	Introduction to NumPy	3	Use NumPy arrays and mathematical functions efficiently.
8	Data Manipulation and Aggregation	3	Apply advanced Pandas functions like GroupBy, merge, and pivoting.
9	Introduction to Statistics	4	Apply descriptive stats and basic inferential techniques (hypothesis testing).
10	Introduction to Machine Learning (ML)	3	Understand ML types, model training/testing using Scikit-learn.
11	Supervised Learning (Linear regression, logistic regression, KNN)	4	Train and evaluate regression and classification models.
12	Unsupervised Learning (K-Means, Hierarchical Clustering, PCA)	3	Apply clustering and dimensionality reduction methods.
13	Capstone Project (end-to-end)	3	Apply all concepts to a real-world dataset and communicate findings.
14	Ethical Considerations in Data Science	1	Understand ethical and responsible data usage, bias, and privacy.
	Total	42	

Course Outcome	Explain the data science workflow and its relevance across disciplines. Use Python libraries like Pandas, NumPy, Matplotlib, and Seaborn for data manipulation and visualization. Clean, pre-process, and explore data for insights. Apply basic statistical concepts and hypothesis testing to support conclusions. Build and evaluate machine learning models using Scikit-learn. Complete a data science capstone project that demonstrates real-world problem solving. Recognize and address ethical concerns such as data bias and privacy issues in data science applications.
-----------------------	---

Learning Resources	1. <i>Python for Data Analysis</i> – Wes McKinney 2. <i>Introduction to Machine Learning with Python</i> – Andreas C. Müller & Sarah Guido 3. <i>Data Science from Scratch</i> – Joel Grus 4. <i>Pattern Recognition and Machine Learning</i> – Christopher M. Bishop 5. <i>The Elements of Statistical Learning</i> – Trevor Hastie, Robert Tibshirani, Jerome Friedman
---------------------------	--

Course Code	IT4129N	Course Name	Intelligent Transportation and Smart Systems	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Communication Systems, Computer Networks</i>	Co-requisite Courses	<i>Mobile Computing, Information and System Security</i>	Progressive Courses	<i>Internet of Things</i>
Course Offering Department		Information Technology		Data Book / Codes/Standards	

Course Objective	1. Understanding ITS Concepts and Architecture 2. Various technological Aspects 3. ITS applications in different transportation modes 4. Understanding vehicle sensors, communication systems, and data processing 5. Gaining knowledge of the operational aspects of ITS lifecycle 6. ITSS threat and security analysis
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1.	Fundamentals of ITS: Overview and history of ITS from both public policy and market economic perspectives, Types of ITS, Historical Background, Benefits of ITS, Overview of ITS Applications.	4	Introduction to intelligent systems, systems and safety standards, modern wireless networks, NR networks and standards
2.	Sensor Technologies and Data Requirements of ITS: Importance of telecommunications in the ITS. Information Management, Traffic Management Centers (TMC). Application of sensors to Traffic management; Traffic flow sensor technologies; Transponders and Communication systems; Data fusion at traffic management centers; Sensor plan and specification requirements; Elements of Vehicle Location and Route Navigation and Guidance concepts; ITS Data collection techniques – Detectors, Automatic Vehicle Location (AVL), Automatic Vehicle Identification (AVI), GIS, video data collection.	10	In depth study of various sensor technologies and their application in transport systems. Physical layer requirements and standards
3.	ITS User Needs and Services and Functional Areas: Introduction, Advanced Traffic Management systems (ATMS), Advanced Traveler Information systems (ATIS), Commercial Vehicle Operations (CVO), Advanced Vehicle Control systems (AVCS), Advanced Public Transportation systems (APTS), Advanced Rural Transportation systems (ARTS).	6	ITS standards and basic components of the system design
4.	ITS Architecture: Regional and Project ITS architecture; Concept of operations; ITS Models and Evaluation Methods; Planning and human factor issues for ITS, Case studies on deployment planning and system design and operation; ITS and safety, ITS and security, ITS as a technology deployment program, research, development and business models, ITS planning.	6	Network layer architecture design and analysis for the intelligent transportation

			building block
5.	ITS Applications: Traffic and incident management systems; ITS and sustainable mobility, travel demand management, electronic toll collection, ITS and road pricing.; Transportation network operations; commercial vehicle operations and intermodal freight; public transportation applications; ITS and regional strategic transportation planning, including regional architectures: ITS and changing transportation institutions Automated Highway Systems- Vehicles in Platoons – Integration of Automated Highway Systems. ITS Programs in the World – Overview of ITS implementations in developed countries, ITS in developing countries.	6	Various case studies
6.	Economics of ITS: Congestion Pricing, Revenue Generation Models.	5	Market survey and demand analysis
7	Security of ITS: Various security model to secure the ITS Systems	5	Security and life cycle of ITSS
	Total	42	

Course Outcome	1. Understanding ITSS Fundamentals 2. ITSS Architecture and Systems Engineering 3. ITSS Standards and Specifications 4. Technological Requirements 5. Emerging Trends in ITSS 6. Impact of ITSS on Safety and Efficiency
-----------------------	---

Learning Resources	Books: 1. Sussman, Joseph. Perspectives on Intelligent Transportation Systems (ITS). New York, NY: Springer, 2010. 2. Mashrur A. Chowdhury, and Adel Sadek, Fundamentals of Intelligent Transportation Systems Planning, Artech House, Inc., 2003. 3. Fundamentals of intelligent transportation systems planning By Mashrur A. Chowdhury, Adel Wadid Sadek. 4. Lawrence A. Klein, Sensor technologies and Data requirements of ITS 5. ITS Hand Book 2000: Recommendations for World Road Association (PIARC) by Kan Paul Chen, John Miles. 6. Sussman, J. M., Perspective on ITS, Artech House Publishers, 2005. 7. National ITS Architecture Documentation, US Department of Transportation, 2007.
---------------------------	---

Course Code	IT4130N	Course Name	Cyber Physical Systems and Security	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Embedded Systems and IoT</i>	Co-requisite Courses		Progressive Courses	
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	The course aims to introduce the foundational concepts, architectures, and applications of Cyber-Physical Systems (CPS), with a focus on integrating computation, communication, and control in safety-critical environments. It also aims to equip students with the skills to identify, analyze, and mitigate security vulnerabilities in CPS, bridging the gap between embedded systems, networking, and cybersecurity.
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Cyber-Physical Systems Topics: • CPS overview & architecture • Physical processes, embedded computation, and networking integration • CPS lifecycle and design challenges	6	– Understand the definition and key characteristics of CPS. – Recognize differences between CPS, IoT, and embedded systems. – Identify CPS application domains (smart grids, healthcare, autonomous vehicles, industrial automation).
2	CPS Modeling and Simulation Topics: • Mathematical modeling of CPS • Hybrid automata and discrete-event systems • Model-based design tools (MATLAB/Simulink, Stateflow, Ptolemy II)	6	– Model CPS using state machines and hybrid systems. – Use simulation tools for CPS verification and validation.
3	Communication and Networking in CPS Topics: • Network architectures for CPS • Time-sensitive networking (TSN) • Wireless sensor networks (WSNs) and industrial protocols (Modbus, OPC UA)	6	– Explain CPS communication requirements and constraints. – Evaluate networking protocols for real-time and safety-critical CPS applications.
4	Security Threats and Vulnerabilities in CPS Topics: • Attack surface of CPS (sensors, actuators, communication links) • Threat modeling in CPS	6	– Classify types of cyber and physical attacks on CPS. – Understand CPS-specific security challenges compared to IT systems.

	Real-world CPS attacks: Stuxnet, BlackEnergy, Triton		
5	CPS Security Mechanisms and Standards Topics: - Authentication, authorization, and access control in CPS - Intrusion detection and prevention for CPS - CPS safety standards (IEC 62443, NIST CPS Framework, ISO 26262)	6	- Design security controls specific to CPS. - Apply relevant CPS security standards and frameworks.
6	Resilience, Fault Tolerance, and Safety in CPS Topics: - Fault detection and diagnosis - Redundancy and fail-safe design - Resilience metrics and recovery strategies	6	- Implement strategies for fault tolerance and resilience. - Evaluate safety-critical performance requirements.
7	Case Studies and Future Trends Topics: - Case studies: Smart grids, autonomous vehicles, industrial control systems - Emerging trends: AI in CPS security, digital twins, quantum-safe CPS security	6	- Analyze real-world CPS deployments and their security measures. - Discuss emerging trends and research directions in CPS security.
	Total	42	

Course Outcome	Upon successful completion, students will be able to: CO1: Explain the architecture, components, and applications of CPS across domains. CO2: Apply modeling and simulation techniques for CPS design and analysis. CO3: Identify potential security threats and vulnerabilities in CPS environments. CO4: Propose and implement strategies to secure CPS from cyber and physical attacks. CO5: Evaluate CPS performance, safety, and resilience in real-world scenarios.
-----------------------	--

Learning Resources	Textbooks: - Edward A. Lee & Sanjit A. Seshia, <i>Introduction to Embedded Systems: A Cyber-Physical Systems Approach</i>, MIT Press, 2nd Edition. - Raj Rajkumar, Insup Lee, Lui Sha, John Stankovic, <i>Cyber-Physical Systems: The Next Computing Revolution</i>, Springer. - Fei Hu, <i>Cyber-Physical Systems: Integrated Computing and Engineering Design</i>, CRC Press. Reference Books & Resources: - NIST Special Publication 1500-201: <i>Framework for Cyber-Physical Systems</i>. - Kim, K., & Kumar, P., <i>Cyber-Physical Systems: Foundations, Principles and Applications</i>, Elsevier. - Wolf, M., <i>Cyber-Physical Systems Security</i>, Springer. - Selected IEEE Transactions on Cyber-Physical Systems, ACM Transactions on Cyber-Physical Systems.
---------------------------	--

Course Code	IT4131N	Course Name	CAD for VLSI	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Introductory course on Digital Logic and Computer Organization</i>	Co-requisite Courses	<i>VLSI Physical Design, Digital System Design</i>	Progressive Courses	<i>Advanced VLSI Physical Design Automation, High - Level Synthesis (HLS)</i>
Course Offering Department		<i>Information Technology</i>		Data Book / Codes/Standards	

Course Objective	Primary objective of this course is to introduce the concepts of Computer Aided Design Algorithms used during the VLSI Design Process
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: VLSI design flow, challenges.	2	Understanding of the complete VLSI design, identification of key challenges such as power, area and analysis of the role of CAD tools.
2	Verilog/VHDL: Introduction and use in synthesis, modeling combinational and sequential logic, writing test benches.	6	learn the fundamentals of HDL, synthesize RTL designs, and exposure on test benches for functional verification.
3	Logic synthesis: two-level and multilevel gate-level optimization tools, state assignment of finite state machines.	4	Understanding of logic synthesis and optimizations
4	Basic concepts of high-level synthesis: partitioning, scheduling, allocation and binding, Technology mapping.	5	Exposure on High level synthesis for hardware implementation
5	Physical design automation: Review of MOS/CMOS fabrication technology, VLSI design styles: full-custom, standard-cell, gate-array and FPGA.	5	Review MOS/CMOS fabrication processes and compare various VLSI design styles
6	Physical design automation algorithms: Floor-planning, placement, routing, compaction, design rule check, power and delay estimation, clock and power	8	learn key physical design automation steps with emphasis on analog and mixed-signal design challenges.

	routing, etc. Special considerations for analog and mixed signal design.		
7	Testability issues: Fault modelling and simulation, test generation, design for testability, built-in self-test. Testing SoCs.	8	Learn the test and design for testability of Integrated Circuits.
8	Basic concepts of verification: Design verification techniques based on simulation, analytical and formal approaches. Functional verification. Timing verification, Formal verification, Basics of equivalence checking and model checking. Hardware emulation.	4	Understand different techniques for design verification of Integrated circuits
	Total	42	

Course Outcome	1. Overall idea of VLSI Design cycle and role of CAD. 2. Different techniques and algorithms used during synthesis, layout design, test and verification process during VLSI design cycle. 3. Acquaintance with the state-of-the-art design automation tools used during CAD based VLSI design and hands-on experience
-----------------------	--

Learning Resources	Text Books: 1. M.D.Ciletti, "Modeling, Synthesis and Rapid Prototyping with the Verilog HDL", Prentice-Hall. 2. M.G.Arnold, "Verilog Digital – Computer Design", Prentice-Hall. 3. VLSI Physical Design Automation Theory and Practice – by Sait, Youssef, World Scientific. 4. Algorithms for VLSI Physical Design Automation by Naveed Shervani, Springer International Edition, 3rd Edition, 2005. 5. G. De Micheli. Synthesis and optimization of digital circuits, 1st edition, 1994 6. Gary D. Hachtel and Fabio Somenzi, Logic Synthesis and Verification Algorithms. Springer. Reference Books: 1. Digital Integrated Circuits- A Design Perspective by J M Rabaey, Prentice Hall, 3rd Edition, 2012. 2. S. Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, Prentice Hall, 2nd edition, 2003. 3. Douglas L. Perry, VHDL: Programming by Example 4th Edition, TMH. 4. D. D. Gajski, N. D. Dutt, A.C.-H. Wu and S.Y.-L. Lin, High-Level Synthesis: Introduction to Chip and System Design, Springer, 1st edition, 1992. 5. Rudiger E bendt, Görschwin Fey, Rolf Drechsler. Advanced BDD Optimization.
---------------------------	---

Course Code	IT4132N	Course Name	Cognitive Radio Networks	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Wireless Communications/Wireless Networks, Commun. Engineering	Co-requisite Courses	Signals and Systems/ DSP, Soft computing	Progressive Courses	5G Beyond, IoT Communications. Cognitive Radio Access Networks (CRANs)
Course Offering Department	<i>Information Technology</i>			Data Book / Codes/Standards	

Course Objective	To enable students to design spectrum and energy efficient sustainable wireless communication system design. Coexistence of Primary and Secondary network operations to offer a large number of wireless nodes connectivity seamlessly. IoT applications specific communication system design and role in wireless network design in 5G and Beyond.
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Next generation/5G wireless networks: Introduction to cognitive radio networks, spectrum scarcity problem, network architectures, Cognitive cycle and functional components	2	1G to 5G evolution to revolution, promises and potential
2	Spectrum sensing (SS) in CRN, Different SS techniques, -energy detection, Matched filter detection, , feature detection, Cyclo-stationary feature detection, likelihood ratio test (LRT), GLRT techniques Cooperative SS- soft and hard decision fusion, Energy Efficient CSS, security threats in CSS. Security threats in CSS	6	Spectrum information, assessment through sensing & security threats
3	Joint SS and Data Transmission, -Link layer design and common control channel, resource allocation-power allocation and channel assignment, optimized system design, conventional and ML/DL methods	4	Spectral and energy efficient communication system, coexistence of primary and secondary networks
4	Multi-hop CRN: Routing protocols, both centralized, and distributed geographic forwarding and probabilistic approaches-outage analysis	4	Relay based multihop CRN, Link reliability analysis
5	Network Protocol Design for CR: Transport layer protocol design, both TCP- and equation-based Standards and applications	4	TCP in CRN, architecture and functionality

6	Security threats in CRN: Security threats in SS- PUEA and SSDF attacks. Eavesdropping and secrecy outage in CRN, Jamming for eavesdropping protection, jammer selection, ergodic capacity analysis	4	Jamming and Eavesdropping impact on Cooperative CRN, outage secrecy analysis
7	Energy Harvesting in CRN-Wireless energy transfer and scavenging, SWIPT concept, linear and non-linear modeling of EH, Circuit design and interfacing RF energy harvesting boards	4	Sustainable CRN, TS, PS, SWIPT mode impact of EH
8	Machine Learning in CRN: Spectrum Prediction and spectrum database formation, RL	6	ML and RL is spectrum management
9.	Application Specific System Design in CRN- Wireless Medical Telemetry Services (WMTS), cognitive radio vehicular networks (CR VANET), CR for emergency communication, CR-IoT, D2D cooperative CRN, CRN for smart city	8	Applications specific CRN architecture and analysis-WMTS and VANET, disaster situation
	Total	42	

Course Outcome	Cognitive networks represent a relatively novel paradigm in which it is supposed that users device sense and understand the electromagnetic environment to become aware of the available transmission opportunities even in frequency intervals nominally assigned to other specific services. When available, cognitive devices may exploit these opportunities provided they vacate the channel as soon as a legitimate user starts transmission. Besides providing potential very high gains in terms of spectrum efficiency, cognitive radios and networks pose several challenges that will be described in the lectures. CO1: The students will get exposure on future wireless communication system design enabling both primary and secondary network operations together without causing much degradation in performance. CO2: Primary Spectrum sensing and prediction CO3: Joint Spectrum Sensing and Secondary Data Transmission CO4: Energy Harvesting and Security issue on CRN CO5: CRN applications-Medical and Vehicular Networks
-----------------------	---

Learning Resources	Books: 1. Principles of Cognitive Radio, EzioBiglieri, Andrea J. Goldsmith, Larry J. Greenstein, H. Vincent Poor, Narayan B. Mandayam, Cambridge University Press, 2013 - Computers - 299 pages. 2. Handbook of Cognitive Radio, Editors: Zhang, Wei (Ed.) Springer 3. Cognitive Wireless Communication Networks, Editors: Hossain, Ekram, Bhargava, Vijay K. (Eds.) Springer
---------------------------	--

Course Code	IT4161N	Course Name	Mobile Computing	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	<i>Communication Systems, Computer organization and architecture, Computer Networks</i>	Co-requisite Courses	<i>Internet of Things, Data Science</i>	Progressive Courses	<i>Cloud Computing and Web Services, Parallel and Distributed Systems</i>
Course Offering Department	Information Technology		Data Book / Codes/Standards		

Course Objective	1. Understanding Mobile Computing Fundamentals: portability, connectivity, and context-awareness 2. Mobile Communication and Networking, resource sharing 3. Modern wireless network design 4. Mobile wireless network protocol design criteria, IP 5. Wireless application specific design 6. Wireless and mobile network security analysis
-------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to mobile computing	2	Introduction to various types of mobility
2	Wireless and Cellular network, channel allocation, multiple access	4	Types and categories of wireless networks
3	1G, 2G, systems, GSM standards, architecture	6	Cellular network standards
4	Location management, Handoffs, Authentication	4	Connection management and resource management
5	2G CDMA, 3G CDMA, 4G standards and advances	4	Advanced mobile cellular network
6	IEEE 802.11 WLAN	4	Wireless LAN protocol design: constraints and throughput analysis
7	Bluetooth, HiperLAN architecture, comparison of wireless technologies	4	Comparison of wireless technologies

8	Mobility adaptation, process migration, mobile IP	4	System protocol design issues and mobility management for wireless networks discussed in the previous modules
9	Mobile Ad-hoc networking. MAC protocols, Routing	4	Routing protocol design for QoS determination and performance optimization
10	Energy-efficient computing, Impact of mobility on algorithms	4	Special types of wireless networks and QoS analysis
11.	Security and authentication protocols for mobile computing	2	Security analysis
Total		42	

Course Outcome	1. Concepts of fundamentals of mobile and wireless networks 2. Wireless communication technologies and their applicabilities in the network 3. Study of advanced cellular network 4. Protocol design for various wireless networks 5. Energy efficient routing protocol design for wireless networks 6. Concepts of security and authentication for mobile network
-----------------------	---

Learning Resources	Books: 1. Fundamentals of Mobile Computing by Pattnaik Mall, PHI 2. Mobile Computing, by Talukder Asoke K. Mcgraw Hill 3. Mobile Computing Third Edition, by RAJ KAMAL, Oxford University Press 4. Mobile Communications, by Jochen Schiller, Second Edition, Pearson Education, 2003.
---------------------------	--

Course Code	IT4171N	Course Name	Information and Systems Security Lab	Course Category	PC	L	T	P
						0	0	3

Pre-requisite Courses	Number Theory	Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology			Data Book / Codes/Standards	

Course Objective	Introduction to traditional and present-day cryptographic techniques
-------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Implementation of private key cryptosystems - Block ciphers: such as additive, multiplicative, affine, playfair, vigenere, hill, rotor ciphers	6	Traditional symmetric key ciphers
2	Cryptanalysis of symmetric key ciphers	6	Attacks against symmetric key ciphers
3	Substitution Permutation Networks (SPN), Modern day block ciphers	6	Introduction to modern day ciphers
4	Modes of block cipher operation: ECB, CBC, CFB, OFB etc.	6	Cipher modes
5	Keyed and Keyless Transposition Ciphers	3	Introduction to Transposition ciphers
6	Stream ciphers	3	Introduction to Stream ciphers
7	Implementation of RSA public key cryptosystem, attacks on RSA	6	Asymmetric key ciphers
7	Cryptographic hash, Digital signature schemes, Ethical Hacking	6	Message authentication
	Total	42	

Course Outcome	- Hands-on implementation of traditional and modern ciphers - Symmetric and asymmetric ciphers - Block ciphers vs. stream ciphers - Message authentication and cryptographic hash implementation - Implementation of digital signature schemes
-----------------------	--

Learning Resources	Reading: 1. Behrouz A. Forouzan and D. Mukhopadhyay - Cryptography & Network Security, McGraw Hill. 2. Douglas R. Stinson, Cryptography: Theory and Practice, Chapman and Hall. 3. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall.
---------------------------	---