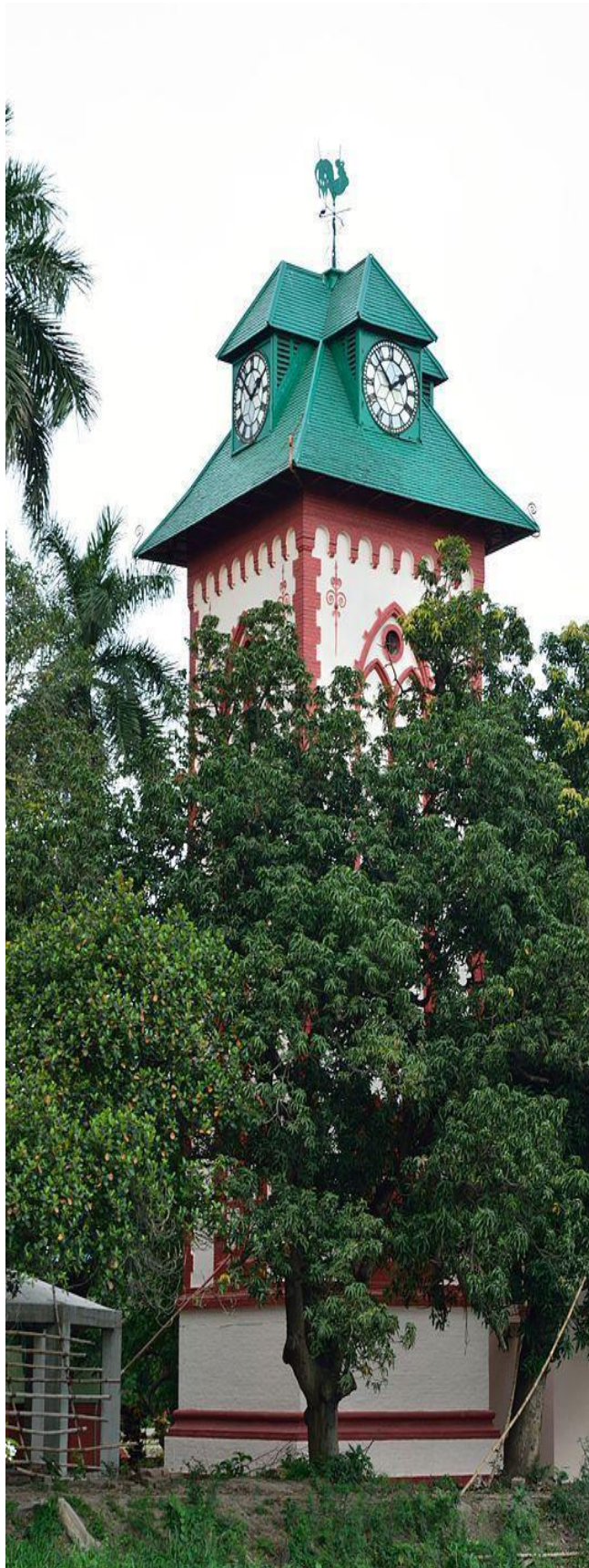




Department of Information Technology

Postgraduate Programmes Course Structure and Syllabus

(Effective from 2025-26 admitting batch onwards)



**Indian Institute of Engineering Science
and Technology (IEST), Shibpur**
Botanic Garden, Howrah

Table of Contents

Table of Contents	2
Course Structure	3
M.Tech in Information Technology	3
M.Tech in Information Technology	5
1st Semester Courses Syllabi	6
2nd Semester Courses Syllabi	31

Course Structure

M.Tech in Information Technology

COURSE STRUCTURE FOR M. 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	PC	Algorithms	IT 5101N	3	0	0	3	3	100
2	PC	Advanced Computer Architectures	IT 5102N	3	0	0	3	3	100
3	PC	Advanced Communication Systems	IT 5103N	3	0	0	3	3	100
4	PSE		IT 512XN	3	0	0	3	3	100
5	OE		IT 516XN	3	0	0	3	3	100
		Theory Sub-total		15	0	0	15	15	500
6	PC	Algorithms Laboratory	IT 5171N	0	0	3	2	3	50
7	PC	Advanced Computer Architecture Laboratory	IT 5172N	0	0	3	2	3	50
8	PC	Advanced Communication Systems Laboratory	IT 5173N	0	0	3	2	3	50
		Practical Sub-total		0	0	9	6	9	150
		First Semester Total		15	0	9	21	24	650

PSE for First Semester		OE for First Semester	
1. Soft Computing Techniques (IT 5121N)		1. Complex Systems and Cellular Automata (IT 5161N)	
2. Embedded Systems and Internet of Things (IT 5122N)		2. Wireless Networks (IT 5162N)	
3. Pervasive Computing (IT 5123N)		3 Cryptographic Techniques (IT 5163N)	
4. Discrete and Computational Geometry (IT 5124N)		4. Medical Image Processing (IT 5164N)	
5. Information and Coding Theory (IT 5125N)		5. Embedded Systems and HW-SW Co-design (IT 5165N)	
6. Design of Operating Systems(IT 5126N)		6. Advance Data Science (IT 5166N)	
7. Mathematics for Computation (IT 5127N)			

Second Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	PC	Advanced Database Management System	IT 5201N	3	0	0	3	3	100
2	PC	Network and Information Security	IT 5202N	3	0	0	3	3	100
3	PC	Internet and Distributed Computing	IT 5203N	3	0	0	3	3	100
4	PSE		IT 522XN	3	0	0	3	3	100
5	OE		IT 526XN	3	0	0	3	3	100
		Theory Sub-total		15	0	0	15	15	500

6	P	M.Tech. project/Term-paper	IT 5291N	0	0	3	2	3	50
7	O	Seminar/Viva Voce	IT 5292N	0	0	3	2	3	50
		Practical Sub-total		0	0	6	4	6	100
		Second Semester Total		15	0	6	19	21	600

PSE for Second Semester				OE for Second Semester			
1.	Multimedia Coding and Compression (IT 5221N)			1.	CAD Algorithms for VLSI (IT 5261N)		
2.	Mobile Computing (IT 5222N)			2.	Computational Topology (IT 5262N)		
3.	Approximation Algorithms (IT 5223N)			3.	IoT Systems (IT 5263N)		
4.	Real Time Systems (IT 5224N)			4.	DSP Algorithms (IT 5264N)		
5.	Machine Learning (IT 5225N)			5.	Cyber-Physical Systems and Security (IT 5265N)		
6.	Cloud and Services Computing (IT 5226N)			6.	Cognitive Radio and Networks (IT 5266N)		
7.	High-Performance and Parallel Computing (IT5227N)			7.	Quantum and Neuromorphic Computing (IT 5267N)		

Third Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	VAC			3	0	0	3	3	100
2	P	M. Tech. Thesis	IT 6191N				12	24	300
3	O	Progress Seminar and Viva-voce	IT 6192N				6		100
4	I	Summer internship (6-8 weeks) evaluation					2		50
		Third Semester Total					23		550

Fourth Semester									
Sl. No.	Type	Course Name	Course code	Class Load/Week			Credit	Class load/ week	Marks
				L	T	P			
1	P	M. Tech. final thesis	IT 6291N				22	30	400
2	O	Thesis Seminar and Viva-voce	IT 6292N				8		200
		Fourth Semester Total					30		600

M.Tech in Information Technology

Detailed Syllabus

1st Semester Courses Syllabi

M. Tech in Information Technology

Course Code	IT5101N	Course Name	Algorithms	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Discrete Mathematics, Programming and Data Structure	Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology			Data Book / Codes/Standards	None

Course Objectives	To introduce the performance of algorithms in terms of time and space complexity To choose the appropriate algorithm design technique for a specified problem To introduce graph algorithms and their efficiency analysis To and understand NP-completeness and randomized algorithms
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Models of Computation; Algorithms and Complexity; Best case, worst case and average case; asymptotic notations	6	Analyze the asymptotic performance of algorithms
2	Elementary Algorithms: Sorting and searching; search trees; balanced trees; hashing	6	Understanding the sorting and searching algorithms with complexity analysis
3	Lower bound theory	2	Introduction to lower bound of an algorithm
4	Optimization problems; Dynamic programming and Greedy method; theoretical foundations of greedy method	6	To solve problems using algorithm design techniques such as the greedy method, divide and conquer, dynamic programming
5	Graph algorithms: BFS and DFS, Minimum Spanning Trees, Shortest Paths, Max Flow	8	Understanding graph traversal algorithms, shortest path problems and analyze their time complexities
6	Randomized algorithms: identity testing, primality and min cut, Number theoretic algorithms	6	motivation of randomized algorithms, expected running time of randomized algorithms
7	Limit of computation, complexity classes: P, NP and NP completeness	8	To understand P and NP classes
	TOTAL	42	

Course Outcome	Understand the algorithm design with time and space complexity Able to analyze the different algorithm design techniques for a given problem Understand the concepts of complexity classes Explain the different ways to analyze randomized algorithms
-----------------------	--

Learning Resources	1. Cormen, Leiserson, Rivest and Stein: Introduction to Algorithms, 3rd Edition (The MIT Press) 2. E. Horowitz and S. Sahni, Fundamental of Computer Algorithms, Galgotia Pub. 3. Aho, Hopcroft and Ullman: The Design and Analysis of Computer Algorithms (Pearson)
---------------------------	--

Course Code	IT 5102N	Course Name	Advanced Computer Architectures	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Basic knowledge of Computer Organization, Digital Logic, and Programming	Co-requisite Courses	- Parallel and Distributed Computing - Operating Systems (Advanced / Real-Time) - Embedded Systems and Hardware/Software Co-Design - Computer System Simulation & Performance Modelling (optional) - Machine Learning Hardware Acceleration (optional)	Progressive Courses	- High-Performance and Parallel Computing (HPC) - Advanced VLSI Design & Microarchitecture Optimization - Network-on-Chip (NoC) & Scalable Interconnects - Reconfigurable and Domain-Specific Architectures - Quantum and Neuromorphic Computing - Operating System Architecture and Virtualization - Compiler Design for High-Performance Architectures
Course Offering Department		Information Technology		Data Book / Codes/ Standards	None

Course Objectives	By the end of the course, students will be able to: - Analyze and compare performance, power, and efficiency of modern processor designs - Understand instruction-level, thread-level, and data-level parallelism - Explore and critique the architecture of modern multi-core and many-core systems - Apply architectural principles to design scalable and efficient computing systems - Understand the role of emerging paradigms in computer architecture (e.g., accelerators, neuromorphic, and quantum)
-------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Module 1: Fundamentals & Quantitative Analysis (Weeks 1–2) Topics: - Review of basic computer architecture concepts - Performance metrics: CPI, MIPS, Amdahl's Law, EDP - Introduction to benchmarking and simulation tools (SPEC, GEM5 overview)	6	- Analyze and interpret architectural performance metrics - Apply Amdahl's and Little's Law to performance analysis - Evaluate system efficiency using benchmark data
2	Module 2: Instruction-Level Parallelism (Weeks 3–4) Topics: - Pipelining: basic concepts, hazards, and forwarding - Branch prediction techniques and speculative execution - Dynamic scheduling: Tomasulo's algorithm, Register renaming - Superscalar and VLIW architectures	6	- Explain how ILP improves performance and its limitations - Model pipeline behavior and evaluate efficiency - Apply Tomasulo's algorithm for dynamic scheduling
3	Module 3: Memory Hierarchy & Optimizations (Weeks 5–6) Topics: - Cache organization: direct-mapped, set-associative - Cache policies: replacement, write, and optimization techniques - Virtual memory, TLB, Page Walk - DRAM, 3D memory, and NVM overview	6	- Design and evaluate memory hierarchy for latency and throughput - Distinguish between different cache strategies and evaluate their impact - Explain virtual memory mechanisms in modern processors
4	Module 4: Thread-Level and Data-Level Parallelism (Weeks 7–8) Topics: Types of multithreading (coarse, fine-grain, SMT)	6	Differentiate between ILP, TLP, and DLP in real-world systems

	– SIMD and Vector Processors – GPU Architecture: CUDA cores, warp scheduling basics – Programming paradigms: CUDA/OpenCL (intro level)		– Describe architectural support for SIMD and GPUs – Assess parallelization potential using TLP/DLP techniques
5	Module 5: Multi-core & Many-core Architectures (Weeks 9–10) Topics: – Overview of multicore processors and CMPs – Cache coherence protocols (MSI, MESI, MOESI) – Memory consistency models – Network-on-Chip (NoC) design and topologies	6	– Explain coherence and consistency challenges in multi-core systems – Apply directory and snooping-based protocols – Analyze trade-offs in NoC design
6	Module 6: Domain-Specific Architectures (Weeks 11–12) Topics: – GPU deep dive (e.g., NVIDIA Ampere architecture) – Tensor Processing Units (TPUs) and AI accelerators – Edge computing and energy-aware architecture – Neuromorphic computing overview	6	– Evaluate architecture trade-offs for specialized domains – Describe architectural adaptations for AI/ML tasks – Compare general-purpose vs domain-specific accelerators
7	Module 7: Emerging Trends and Research Directions (Week 13) Topics: – RISC-V: instruction set and design philosophy – Reconfigurable computing: FPGAs, CGRAs – Quantum computing architectures: basics and current models – Reliability, fault-tolerance, and secure computing	3	– Identify emerging paradigms and their architectural impact – Explain how reconfigurable hardware enhances flexibility – Assess architectural solutions for security and fault tolerance
8	Module 8: Review, Research Presentation, and Evaluation (Week 14) Topics: – Review of major concepts across modules – Student presentations on selected papers / project work – Q&A and discussions on modern real-world architectures (e.g., Apple M3, AMD EPYC) – Final examination preparation	3	– Integrate and apply knowledge of all architectural layers – Present and critique modern architectural innovations – Demonstrate understanding through mini-projects and presentations
6	TOTAL	42	

Course Outcome	CO1. Analyze the performance of computer systems using quantitative models such as Amdahl's Law and benchmarking tools. CO2. Demonstrate a comprehensive understanding of instruction-level parallelism (ILP) through pipelining, dynamic scheduling, and speculative execution. CO3. Evaluate memory hierarchy design decisions, including caching strategies, virtual memory, and emerging memory technologies. CO4. Differentiate and apply various levels of parallelism (ILP, TLP, DLP) in the context of multi-core processors, GPUs, and vector architectures. CO5. Design coherent multi-core systems considering cache coherence protocols, memory consistency models, and Network-on-Chip (NoC) topologies. CO6. Assess the suitability and architecture of domain-specific processors like TPUs, FPGAs, and neuromorphic chips for targeted applications. CO7. Investigate emerging trends such as RISC-V, reconfigurable computing, and quantum architecture, and critique their potential impact. CO8. Communicate and present architectural innovations through technical reports and oral presentations, demonstrating clarity and depth of understanding.
-----------------------	---

Learning Resources	Primary Textbooks • Computer Architecture: A Quantitative Approach – John L. Hennessy and David A. Patterson • Computer Organization and Design: The Hardware/Software Interface – Patterson and Hennessy • Advanced Computer Architecture: Parallelism, Scalability and programmability – Kai Hwang, Naresh Jotwani Supplementary References • Parallel Computer Architecture – David E. Culler and Jaswinder Pal Singh • Modern Processor Design – John Shen and Mikko Lipasti • Research Papers from IEEE Micro, ACM TACO, ISCA, HPCA
---------------------------	---

Course Code	IT 5103N	Course Name	Advanced Communication Systems	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Students are expected to have knowledge on Signals and Systems, Random Processes, and Communication Engineering.	Co-requisite Courses	Communication Laboratory, Information and Coding Theory	Progressive Courses	Communication Laboratory, Information and Coding Theory Wireless Communication, Computer Networks and Data Communication
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Introduction to signals, spectrum, bandwidth in communication. Familiarize students with different baseband and carrier modulated data transmission. Transmission bandwidth and probability of error calculation, improved receiver design. Data transmission for jamming protection, low probability of intercept, parameter estimation, hypothesis testing
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Baseband, narrowband and wideband signals and noise representation and characteristics of communication channels.	4	Introduction to signals, spectrum, bandwidth, noise and important components of communication system
2	Baseband binary signal transmission over band limited channels, Transmission coding and PSD, ISI and its control	4	Transmission of analog signal as digital form, different signalling techniques, bandwidth calculation, and channel impairments on data rate
3	Synchronization techniques: Carrier, bit and frame synchronization, optimum receive filtering, match filtering.	6	Baseband receivers and noise performance, optimal

			performance for minimum probability of bit error
4	M-ary signals orthogonal representation, Gram-Schmidt procedure, signal space concept, Bandwidth efficient	4	Signal space representation, orthogonalizations for interference mitigation
5	Digital carrier modulation techniques: Binary and M-ary shift keying techniques, QPSK, MPSK, MSK, GMSK, coherent and non-coherent detection, PSD and bit error rate calculation	10	Different digital data transmission techniques through free-space, transmission bandwidth requirements and error performance analysis
6	Spread Spectrum: Concept of spectrum spreading, process gain, properties and generation of code patterns, DSSS, FHSS, THSS techniques and their comparison	8	Impact of spectrum spreading on data transmission over hostile channel, multiple access, jamming protection
7	Principle of detection and estimation: Binary and M-ary hypothesis testing. Bayes' likelihood ratio test, waveform estimation, linear estimation problems, Wiener filtering, Kalman filtering	6	Parameter estimation, binary receiver detection, binary hypothesis analysis, detection (classifier) performance analysis
	TOTAL	42	

Course Outcome	By the end of the course, the students will be able to: CO1: Explain the key concepts of baseband and bandpass digital data transmission, transmission bandwidth calculation and relation with data transmission rate CO2: Digital data multiplexing, signalling and co-channel transmission impairment analysis CO3: Improved receiver design for low probability of error in detection. CO4: Data transmission for jamming protection, multipath protection, multiple access scheme CO5: Statistical signal analysis for parameter estimation, hypothesis testing in binary data transmission
-----------------------	--

Learning Resources	1. Digital communications by Simon Haykin-Wiley. 2. Modern Communications and Spread Spectrum by G. R. Cooper and C. D. McGillem-McGraw-Hill Int. 3. Digital Communication Techniques Signal Design and Detection by Marvin K. Simon, Sami M. Hinedi, William C. Lindsey, PHI 4. Digital Communication by J. R. Barry, E. A. Lee and D. G. Messerschmitt, Springer International.
---------------------------	--

Course Code	IT 5121N	Course Name	Soft Computing Techniques	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Discrete mathematics, Calculus: Integration and Differentiation, Concepts of Algorithm and Programming.	Co-requisite Courses	Algorithms	Progressive Courses	Machine learning
Course Offering Department	Information Technology			Data Book / Codes/Standards	None

Course Objectives	• Introduce the concept and need for soft computing as an alternative to traditional (hard) computing methods. • Explain the fundamental principles of fuzzy logic, neural networks, meta-heuristic approaches, and rough set theory. • Explore the integration and hybridization of different soft computing techniques. • Develop problem-solving skills using soft computing approaches for real-world applications involving uncertainty, approximation, and imprecision. • Equip students with the ability to design intelligent systems for optimization, classification, and decision-making tasks.
--------------------------	--

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, Rough Set Theory; Hybrid Systems	4	Understand the necessity of soft computing techniques, distinguishes hard and soft computing paradigms and real-world applications where soft

			computing excels.
2	Fuzzy Logic System: --Fuzzy Set Theory --Fuzzy Relation --Fuzzy Logic and Approximate Reasoning --Fuzzy logic system design, --Applications	12	Understandi ng fuzzy rule-based system design for approximate reasoning for decision making.
3	Artificial Neural Network (ANN): --Basic electrical model of artificial neuron. --NN Architectures: Single Layer feed forward, Multiple layer feed forward and Recurrent network. --Learning Processes: Error correction, memory based, competitive learning, Hebbian learning and Boltzman learning. --Learning Single layer perceptron and <i>Backpropagation Learning</i> --Hopfield NN and Associative Memory --SOM Models and related algorithms. --Applications	12	Understandi ng the basic ANN structures and their training and applications
4	Genetic Algorithm: Difference between Traditional Algorithms and GA, Encoding, Fitness Function, Reproduction, Cross Over, Mutation, Applications.	4	Implementin g GA for real-world optimization tasks
5	Rough Set Theory: Indiscernibility Relations Reducts Rough Approximation Applications.	4	Understandi ng attribute reduction and rule extraction using rough set methods and its application in feature selection and knowledge representati on
6	Meta-heuristics PSO, Ant Colony Optimization, Honey Bee etc.	4	Understands the different meta-heurist ic algorithms and their modeling to solve different real-world optimization problems

	TOTAL	42	
--	--------------	----	--

Course Outcome	• Understand and explain the basic concepts and components of soft computing. • Apply fuzzy logic principles to handle uncertainty and design rule-based systems. • Design, train, and test artificial neural networks for prediction and classification problems. • Implement genetic algorithms to solve optimization problems efficiently. • Use rough set theory for data analysis, feature selection, and decision support. • Integrate multiple soft computing techniques to develop intelligent hybrid systems. • Analyze and evaluate the performance of soft computing solutions in practical applications.
-----------------------	--

Learning Resources	Text Books: 1. Neural networks A comprehensive foundation, Simon Haykin, Pearson Education 2nd Edition 2004 2. Neural Fuzzy Systems- A Neuro-Fuzzy Synergism to Intelligent System, C.T. Lin and George Lee, Prentice Hall 3. Genetic Algorithms in Search, Optimization and Machine Learning, David E. Goldberg. Reference Books: 1. Artificial neural networks by B. Vegnanarayana Prentice Hall of India P Ltd 2005 2. Neural networks in Computer intelligence, Li Min Fu TMH 2003 3. Neural networks, James A Freeman David M S Kapura Pearson Education 4. An Introduction to Genetic Algorithms, Melanie Mitchell, MIT Press. 4. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Jerry M. Mendel. 5. Fuzzy Logic with Engineering Applications, Timothy. J. Ross 6. Fuzzy Sets and Fuzzy Logic – Theory and Applications, G. J. Klir and Bo Yuan, Prentice Hall India
---------------------------	--

Course Code	IT 5122N	Course Name	Embedded Systems and Internet of Things	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	– Computer Architecture & Organization – Digital Logic Design – Microprocessors & Microcontrollers – Programming in C/C++ – Basic Data Communication & Networking	Co-requisite Courses	(Not mandatory but strongly recommended to be taken concurrently or just prior) – Real-Time Systems – to complement RTOS concepts – Sensor Networks – deeper understanding of wireless sensor-based design – Cyber-Physical Systems – for system-level IoT design and modeling – Operating Systems (Advanced) – deeper process/thread/interrupt/scheduling models	Progressive Courses	– Advanced Cyber-Physical Systems (CPS) – Edge Computing and Fog Computing – AI/ML for Embedded Systems (TinyML) – Embedded Linux Systems – Secure Embedded Systems – VLSI Design for Embedded Applications – IoT Application Development – Wireless and Mobile Networks
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	– Understand the architecture, design, and programming of embedded systems. – Explore IoT protocols, communication models, and device integration. – Design, simulate, and analyze real-world applications in embedded and IoT domains. – Evaluate system-level trade-offs in embedded and networked systems. – Develop secure and scalable embedded + IoT systems using modern tools.
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Embedded Systems – Definition & characteristics – Application domains – Embedded vs general-purpose systems	3	Understand the key characteristics and categories of embedded systems.
2	Embedded System Architecture – Microcontrollers vs microprocessors – SoCs, DSPs, FPGAs – Von Neumann vs Harvard architectures	3	Identify and distinguish between hardware components of embedded systems.
3	Embedded Programming Concepts – C for Embedded Systems – Interrupts, timers, memory-mapped I/O – GPIO, ADC/DAC basics	3	Apply embedded C concepts to program microcontrollers.
4	Real-Time Operating Systems (RTOS) – Multitasking, context switching – Scheduling algorithms – RTOS APIs: FreeRTOS/RTS	3	Design real-time embedded applications using RTOS principles.
5	Peripherals and Interfacing – I2C, SPI, UART – Sensor & actuator interfacing	3	Interface peripherals with microcontrollers and optimize for energy efficiency.

	– Power management techniques		
6	IoT Fundamentals – Evolution & architecture of IoT – Enabling technologies (RFID, WSN, MQTT, BLE) – Cloud integration	3	Understand IoT layers, ecosystem, and enabling technologies.
7	Embedded IoT Protocols – MQTT, CoAP, HTTP, 6LoWPAN – LoRa, Zigbee, BLE, NB-IoT – Edge vs cloud communication	3	Evaluate protocol suitability for diverse IoT scenarios.
8	IoT System Design – IoT device lifecycle – Firmware design – Power-aware communication and storage	3	Design firmware and architecture for efficient IoT solutions.
9	Security in Embedded and IoT Systems – Threats & vulnerabilities – Secure boot, authentication, encryption – Blockchain for IoT	3	Analyze security risks and propose protection strategies for IoT.
10	Data Handling in IoT – Data acquisition and preprocessing – Edge AI concepts – Real-time analytics and visualization	3	Integrate data analytics and edge computing into IoT applications.
11	Cloud-IoT Integration – RESTful APIs – Google Firebase, AWS IoT Core, Azure IoT – Remote monitoring dashboards	3	Develop cloud-based control and monitoring of embedded systems.
12	Case Studies & Applications – Smart cities, healthcare, agriculture, energy – Autonomous systems and robotics – Ethics and sustainability	3	Assess real-world systems and ethical implications in IoT deployment.
13	Project Planning and Proposal – Research methods – Problem identification and benchmarking – Proposal writing	3	Prepare a mini-project proposal with literature support.
14	Project Presentations & Future Trends – Student project demos – Emerging trends (TinyML, swarm IoT, neuromorphic computing)	3	Demonstrate project work and identify next-gen trends.
	TOTAL	42	

Course Outcome	By the end of the course, students will be able to: CO1: Analyze embedded system components, design models, and architectures. CO2: Design and implement embedded software using real-time operating systems. CO3: Evaluate IoT protocols and architectures for smart applications. CO4: Develop secure, power-efficient, and connected IoT systems. CO5: Integrate cloud services and data analytics with IoT devices. CO6: Critically analyze case studies and research trends in Embedded IoT.
-----------------------	---

Learning Resources	Core Text Books: 1. Frank Vahid / Tony Givargis, Embedded System Design A Unified Hardware/ Software Introduction – Wiley India 2. Raj Kamal, Embedded Systems: Architecture, Programming and Design, McGraw Hill, 3rd Edition, 2017. 3. Jonathan W. Valvano, Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers, CreateSpace, 3rd Edition, 2018. 4. Arshdeep Bahga & Vijay Madisetti, Internet of Things: A Hands-on Approach, Universities Press, 2015.
---------------------------	--

5. Muhammad Ali Mazidi, The 8051 Microcontroller and Embedded Systems: Using Assembly and C, Pearson, 2nd Edition, 2007.

Reference Books:

1. Peter Marwedel, Embedded System Design, Springer, 2nd Edition, 2010.
2. Jan Axelson, Embedded Ethernet and Internet Complete, Lakeview Research, 2nd Edition.
3. Dieter Uckelmann et al., Architecting the Internet of Things, Springer, 2011.
4. Adrian McEwen & Hakim Cassimally, Designing the Internet of Things, Wiley, 2013.
5. Introduction to Embedded Systems (2e) – by Shibu K V, McGraw Hill Education (India)
6. Computer Organization and Embedded Systems – by Carl Hamacher et al, McGraw Hill International
7. Embedded Systems – Concepts, Design and Programming - by Dave and Dave, Pearson
8. Embedded/Real Time Systems: Concepts, Design and Programming – by Prasad, Dreamtech Press
9. Embedded Microcontrollers and Processor Design – by Osborn, Pearson
10. Embedded Systems Architecture – by Noergaard, ELSEVIER
11. Embedded System Design for students – by Verma, SPD
12. Designing Embedded Hardware – by Catsoulis, SPD, O'Reilly

Supplementary and Online Reading:

1. FreeRTOS official documentation: <https://freertos.org>
2. ARM Developer guides and Cortex-M documentation
3. IEEE IoT Journal, ACM Transactions on Embedded Computing Systems (TECS)
4. Online courses: MIT OpenCourseWare on Embedded Systems, edX IoT Professional Certificate (Berkeley, Curtin, or Harvard)
5. IoT Analytics and Postscapes for industry insights

Course Code	IT 5123N	Course Name	Pervasive Computing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Basic concepts of Computer Networks and Operating Systems	Co-requisite Courses	Embedded Systems, Human-Computer Interaction (HCI), Wireless Communication, Operating Systems Lab.	Progressive Courses	Cyber-Physical Systems, Context-Aware Computing, Wearable Computing, Ambient Intelligence, IoT Systems.
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Primary objective of this course is to introduce the concepts of necessity, characteristics, design, and detailed working principles of an embedded system dominating each electronic and computing sector in industry.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to pervasive computing Introduction to ubiquitous computing, modelling the property, system environment, architecture design	3	Understand foundational concepts of pervasive computing and architecture modeling.
2	Application and requirements Introduction, discussion on pervasive computing research projects, application in every day life and in industry.	6	Explore real-world applications and historical projects in pervasive computing.
3	Smart devices for pervasive computing Introduction to smart devices; mobile smart device users, resources and programming; operating system for pervasive computing	6	Identify and describe mobile smart devices, their OS, and programming approaches.
4	Human computer interaction User interface and its interaction, human centered design, hidden UI via smart, wearable and implanted device, user model acquisition and representation	4	Understand UI design in smart systems including wearable/implantable interfaces.
5	Tagging sensing and controlling Tagging the physical world, sensor and its network, embedded and real time system,	6	Analyze physical tagging, sensors, and real-time embedded systems.
6	Context aware system	4	Model user context including mobility, spatial and temporal awareness.

	Modelling of context aware system, mobility awareness, spatial awareness, temporal awareness		
7	Intelligent systems Basic concept, architecture, design, and systems operation	4	Understand design and operation of intelligent pervasive systems.
8	Pervasive communication Introduction, data network, wireless network, ubiquitous network, future network and issues	8	Study communication layers: data, wireless, and future ubiquitous networks.
9	Introduction to pervasive computing Introduction to ubiquitous computing, modelling the property, system environment, architecture design	4	Reinforce understanding of environment modeling and system architecture.
	TOTAL	42	

Course Outcome	After completion of this course the student will learn about the basics of modern smart computing, sensing system, context aware and intelligent system, HCI, and pervasive communication
-----------------------	---

Learning Resources	1. Ubiquitous Computing: Smart Devices, Environments and Interactions, By Stefan Poslad, J Wiley. 2. Fundamentals of Mobile And Pervasive Computing, By Adelstein, Tata McGraw-Hill Education 3. Pervasive Computing and Networking edited by Mohammad S. Obaidat, Mieso Denko, Isaac Woungang, J Wiley 4. Pervasive Computing, by Jochen Burkhardt, Dr. Horst Henn, Stefan Hepper, Klaus Rindtorff, Thomas Schack, Pearson Education India 5. Pervasive Computing: Concepts, Technologies and Applications, By Minyi Guo, Jingyu Zhou, Feilong Tang, Yao Shen, CRC
---------------------------	---

Course Code	IT 5124N	Course Name	Discrete and Computational Geometry	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Concepts of Computer Graphics, Data Structures and Algorithms	Co-requisite Courses	Discrete Mathematics, Algorithms	Progressive Courses	Advanced Algorithms, Applications of Computational Geometry
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	(a) Introduce rigorous algorithmic analysis for problems in Computational Geometry. (b) Discuss applications of Computational Geometry to graphical rendering. (c) Introduce the notions of Voronoi diagrams and Delaunay Triangulations.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Computational Geometry: Introduction, degeneracy and robustness, Application domains	3	Overview of computational Geometry and its applications
2	Orthogonal range searching (in brief): kd-tree, range tree, Lower Bounds on Algebraic tree model and Geometric data structures (DCEL)	5	Understanding the different tree data structure and how a graph is represented.
3	The Maximal Points Problem (closest pair and farthest pair), Geometric searching, Slab method, Range searching	3	Detailed understanding of the different graphsearching algorithms
4	Point Location and Triangulation, triangulating monotone polygon	4	The idea of graph decomposition and locating a point in a graph region
5	Convex Hull, Different Paradigms, Voronoi Diagram and Delaunay Triangulation, and Quickhull	6	The overall structural view and decomposition of a graph
6	Line segment intersection, Linear programming, Intersection of convex polygons, planes	6	Understanding the methods to deal with more than polygons
7	Clustering Point Sets using Quadrees and Applications	3	Given point sets how the clustering can be done and the corresponding applications
8	Medial Axis, Straight Skeleton, Minkowski Sums	3	Understanding the structure of a graph
9	Shortest Paths and Geodesics	3	Shortest paths between different points and the concept of geodesics

10	Intersection geometry and empty space recognition	3	Finding the empty space
11	Some applications and case studies	3	Applying the learnt algorithms
	TOTAL	42	

Course Outcome	(a) Analyze randomized algorithms for small domain problems. (b) Use line-point duality to develop efficient algorithms. (c) Apply geometric techniques to real-world problems in graphics.
-----------------------	---

Learning Resources	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. 4. Matoušek, Jiří. Lectures on discrete geometry. Vol. 108. New York: Springer, 2002.
---------------------------	--

Course Code	IT 5125N	Course Name	Information and Coding Theory	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Students are expected to have knowledge on Probability, Statistics, random process, and Communication Engineering.	Co-requisite Courses	Advanced Communication Systems	Progressive Courses	Network and Information Security, Mobile Computing, Embedded System Security
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	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	Sources-memoryless and Markov; Information; Entropy for discrete ensembles; Shannon's noiseless coding theorem; source coding; Mutual Information; channel capacity; BSC and other channel	6	Quantification of information
2	Shannon's noisy coding theorem and converse for discrete channels; Continuous channels	6	Noise analysis in communication channels
3	Channel Coding, linear block codes; cyclic codes, Golay codes and cyclic Redundancy Check (CRC) codes; BCH and Reed-Solomon codes, LDPC codes; Convolution codes; majority logic decoding; Viterbi decoding algorithm	14	Encoding and decoding for digital data transmission
4	Space-time coding for SISO and MISO systems	6	Wireless channel model and data coding for transmission
5	Information theory and coding for physical layer security: Wire-tap channel, Slowly fading wiretap channel, Relay channel and in presence of un-trusted relay	10	Mathematical model and techniques for secured data transmission in noisy channel

	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 coding for wireless networks equipped with advanced systems Security analysis of noisy transmission channel
-----------------------	--

Learning Resources	1. Information Theory, Coding and Cryptography by Ranjan Bose, TMH 2. Elements of Information Theory, 2nd Edition Thomas M. Cover, Joy A. Thomas, Wiley 3. An Introduction to information theory by REZA, FAZLOLLAH M. 4. Error Control Coding from theory to practice by Peter Sweeney John Wiley
---------------------------	---

Course Code	IT 5126N	Course Name	Design of Operating Systems	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Operating Systems, Algorithms	Co-requisite Courses	Computer architectures	Progressive Courses	Network operating systems
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	This course aims towards providing a detailed understanding of structure, purpose and functions of modern operating systems, including hardware and software concepts related to design of OS, process control, concurrency control, processor/disk scheduling, memory management, storage management. The course will include case studies on Linux and Windows operating systems.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Operating Systems: Review of operating systems evolution, operating systems structure, user perspective, introduction to kernel, system calls, system programs, interrupts, context switching	3	Basics of operating systems
2	Hardware and Software Concepts: Hardware components, caching, direct memory access (DMA), hardware support for operating systems, bootstrapping, buffering, software overview, firmware	3	Fundamentals about the hardwares and related softwares
3	Process/Thread Concepts: 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	3	The way a job is treated by the operating system and the interaction between these jobs
4	Concurrency Control: The critical section problem, Synchronization hardware, Semaphores, Classical problems of synchronization, Tools and constructs for concurrency, Detection prevention and avoidance of deadlocks, Recovery from deadlocks, Dynamic resource allocation	4	The understanding of the concurrency mechanisms and resource allocations
5	Processor Scheduling: Scheduling criteria, Pre-emptive Non-pre-emptive scheduling, Priorities, Aging, Scheduling algorithms, Multiple processor scheduling, Algorithm evaluation	3	The ways in which the resources like CPU are shared and scheduled
6	Memory Management: Physical and Virtual Memory, Paging, Segmentation, Locality, Demand Paging, Process creation, Page replacement, Frame allocation, Thrashing	6	Understanding the detailed idea about the memory management
7	Disk Performance Optimization: Evolution of secondary storage, disk scheduling criteria, disk scheduling algorithms, rotational optimization, caching and buffering, RAID	3	The disk management and the issues with achieving better

			performance of disk writes and reads
8	File Management: File Systems, File Organization, File Allocation, Free Space Management, Swap Space Management, File Access Control	4	The accessibility of the stored data and the corresponding algorithms
9	I/O Management: I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O Requests to Hardware Operations	5	Understanding about the methods achieve high I/O efficiency
10	Case Studies: Design of UNIX operating system: Architecture of Unix OS, kernel, structure of buffer cache, buffer pool, file system, process control, I/O system	5	Understanding the existing operating systems
11	Design of Windows operating systems: Design Principles, Architecture, System Components, Environmental Subsystems, File system, Programmer Interface	5	Understand the existing operating systems
	TOTAL	42	

Course Outcome	1. Detailed understanding of structure, purpose and functioning of modern operating systems 2. Problem solving capabilities related to process management, scheduling, synchronization, concurrency control, deadlock handling 3. Understand memory management, use of virtual memory in modern operating systems, and the structure of the most common file-systems 4. Understanding of Linux and Windows operating systems
-----------------------	---

Learning Resources	Essential Reading: 1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, Wiley. 2. Harvey M. Deitel, Paul J. Deitel and David R. Choffnes. Operating systems. Delhi.: Pearson Education: Dorling Kindersley. 3. Maurice J. Bach, The design of the UNIX operating system. Englewood Cliffs, NJ: Prentice-Hall. 4. Arpaci-Dusseau, Remzi H., and Andrea C. Arpaci-Dusseau. Operating systems: Three easy pieces. Vol. 1. Arpaci-Dusseau Books, 2015. References: 1. William Stallings, Operating Systems, Internals and Design Principles, Pearson Education. 2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, Prentice Hall 3. Randal E. Bryant and David Richard O'Hallaron. Computer systems: a programmer's perspective, Prentice Hall
---------------------------	---

Course Code	IT 5127N	Course Name	Mathematics for Computation	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Basics of Discrete Mathematics, Programming and Data Structure	Co-requisite Courses		Progressive Courses	Cryptography and Security, Machine Learning
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	• Develop a strong foundation in linear algebra, calculus, probability, and discrete mathematics. • Enable students to apply mathematical reasoning in algorithm analysis, data structures, and programming logic. • Provide tools for modeling and analyzing problems in computation and data-driven applications. • Strengthen students' ability to translate mathematical ideas into computational frameworks.
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Review of Group Theory, Rings and Fields	6	understand and analyze algebraic structures their applications in symmetry and transformations.
2	Queuing Theory: Introduction, Analysis of a single queue, Queuing Network, Operational Loss	8	Classify different types of queuing models and understand their assumptions and applications.
3	Vector Calculus: Surface and Volume Integral, Stoke's Theorem, Green's Theorem, Divergence Theorem	8	Understand and apply the concepts in analyzing physical and geometric phenomena.
4	Probability and Statistics Basics of Covariance and Correlation, Conditional probability and Bayes Theorem, Probability distribution functions for univariate and multivariate variables, Maximum Likelihood estimation, Maximum a posteriori estimation, Random Process, Stochastic Process and Modelling	10	Understanding theory for data analysis, machine learning, and decision-making under uncertainty.
5	Linear Algebra: Matrix Decomposition, Matrix Diagonalization and its Applications, Vector Space, Tensor algebra and tensor analysis	10	understand and apply the concepts of vectors, matrices, linear transformations, and systems of linear equations for solving mathematical and real-world problems.
	TOTAL	42	

Course Outcome	• Understand and apply concepts of mathematical models like queuing theory to apply solving computing and data-driven applications. • Apply discrete mathematics concepts including logic, group theory, ring and field theory to computation and its application in security and other areas. • Utilize probability and statistics for data analysis, machine learning, and decision-making under uncertainty. • Use calculus for analyzing change and modeling in algorithm design.
-----------------------	--

Learning Resources	1. Lehman, Eric, Tom Leighton, and Albert R. Meyer. Mathematics for computer science. Technical report, 2006. Lecture notes, 2010. 2. Kleinrock, Leonard. "Queueing Systems: Volume 1: Theory, 1975." A Wiley-Interscience Publication (1975). 3. Kleinrock, Leonard. Queueing systems, volume 2: Computer applications. Vol. 66. New York: wiley, 1976. 4. Gilbert and Nicholson: Modern algebra with applications. Second Edition. 5. S Barry Cooper: Computability Theory 6. Angelo Margaris: First order mathematical logic.
---------------------------	---

Course Code	IT 5161N	Course Name	Complex Systems and Cellular Automata	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Basics of Discrete Mathematics, Programming and Data Structure	Co-requisite Courses		Progressive Courses	Master Degree Projects
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	1. To learn an alternative model of computation 2. To understand the nature-inspired computing and computation in nature 3. To understand the cellular automaton as a modeling tool and a tool of technology development
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Function Composition, Parallel Map, Computable Functions, Finite Automata, Turing Machine	4	Recapping Foundational Concepts
2	Injectivity and Surjectivity of cellular automata, Garden-of-Eden theorem, balance property	6	Stepping into Cellular Automata
3	Reversible cellular automata, de Bruijn graphs and testing reversibility and surjectivity of one-dimensional CA, Amoroso and Patt's algorithm, Reachability tree	8	Understanding Reversibility
4	Models of Computation, Computation in Nature, Cellular Automata as Natural Models of Computation, Computational universality in cellular automata	6	Understanding Computation in Nature
5	Introduction to dynamical systems, cellular automata as discrete dynamical systems, limit sets and attractors, Undecidability	4	Understanding Chaos
6	Chaos, sensitivity and mixing properties, expansivity, Lyapunov Exponent, Modelling of physical systems, Understanding biological self-reproduction using cellular automata	4	Understanding Chaos
7	Uniformity in cellular automata, introduction of non-uniformity, hybrid, asynchronous and network cellular automata	4	Understanding Role of Non-uniformity
8	Cellular automata as technology	4	Enabling in Developing Technology with Cellular Automata
	TOTAL	42	

Course Outcome	1. Learning cellular automaton as an alternative model of computation 2. Understanding the nature-inspired computing and computation in nature 3. Understanding cellular automaton as a modeling tool and enabling in developing cellular automata based technology
-----------------------	---

Learning Resources	1. Joel L. Schife. <i>Cellular Automata: A Discrete View of the World</i>. Wiley-Interscience 2. Tommaso Toffoli, Norman Margolus. <i>Cellular Automata Machines: A New Environment for Modeling</i>. MIT Press
---------------------------	---

Course Code	IT 5162N	Course Name	Wireless Networks	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Students should have knowledge in communication technologies and computer networking	Co-requisite Courses	Advanced Communication Systems	Progressive Courses	Internet and Distributed Computing, Network and Information Security
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Wireless medium, channel modelling	2	
2	Wireless physical layer alternatives for networks: Carrier modulation, broadband communication and spread spectrum	3	Modulation scheme design
3	Concept of antennas for radio transmission, path loss	3	Types of wireless communication channels
4	Spectrum access: Cellular system, TDMA, FDMA, CDMA, (C/I) ratio	4	Wireless communication architecture
5	Wireless network operation: Mobility management, radio resource and power management	3	Wireless resource management
6	Advanced air-interface concepts: Radio frequency analysis, OFDM, SISO MIMO concept, LTE and LTE advanced	4	Modern wireless network design
7	Data network: GPRS and higher data rate, short messaging service	4	Application specific protocol design
8	Wireless LAN: Standards – Architecture – Services – physical layer and MAC sublayer management, IEEE 802.11a, b, g	4	Protocol design for various wireless networks
9	Wireless HIPERLAN	3	-do-
10	Mobile Ad hoc Networks- WiFi and WiMAX - Wireless Local Loop	4	-do-
11	Wireless PAN: Bluetooth technology	4	-do-
12	Modern wireless technologies: SDR, WRAN, LoRa	4	Software defined network design
	TOTAL	42	

Course Outcome	This course covers advanced topics in wireless networking and mobile computing, including supporting wireless technologies, various types of wireless networks, mobile protocols, and emerging wireless and mobile technologies.
-----------------------	--

Learning Resources	1. William Stallings, "Wireless Communications and Networks", Pearson Education, 2002 2. Jochen Schiller, "Mobile Communications", Second Edition, Pearson Education, 2003. 3. Cory Beard and William Stallings, "Wireless Communication Networks and Systems", 1 st edition 4. Kaveh Pahlavan, Prasanth Krishnamoorthy, "Principles of Wireless Networks", First Edition, Pearson Education, 2003 5. C.K.Toh, "AdHoc Mobile Wireless Networks", First Edition, Pearson Education, 2002.
---------------------------	---

Course Code	IT 5163N	Course Name	Cryptographic Techniques	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Number Theory, Discrete Mathematics	Co-requisite Courses		Progressive Courses	Network and Information Security
Course Offering Department	Information Technology		Data Book / Codes/Standards	None	

Course Objectives	This course aims towards teaching the basics of computer/data security goals and techniques. This will help the students to develop a mathematical basis of cryptography and cryptanalysis.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Basics of Security and Cryptography: 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	Basic about cryptography
2	Mathematical Background: Introduction to Number theory, Modular arithmetic, Prime number generation, Primality Testing, Euclidean Algorithm, Chinese Remainder Theorem, Fermat's Little Theorem and Euler's Theorem, Index of Coincidence	3	Mathematical foundation required for cryptographic algorithms
3	Cryptographic Techniques: Introduction to Substitution Ciphers, Transposition Ciphers, Encryption and Decryption, Symmetric and Asymmetric Key Cryptography, Key Range and Key Size, Key Domain Analysis	4	Basic about different cryptographic techniques
4	Traditional Private Key Cryptography: Symmetric Encryption. Definitions. Chosen-Plaintext Attack. Chosen-Ciphertext Attack, Known-Plaintext Attack. Known-Ciphertext Attack, Pattern Analysis Attacks, Statistical Attacks. Block ciphers and Stream ciphers	4	Knowledge about symmetric encryption and different attacks
5	Modern Private Key Cryptosystems: Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, Variations on DES - RC4, RC5, Modes of operation of block ciphers, Key stream generation in Stream ciphers	6	Modern symmetric encryption systems

	Public Key Cryptosystems: RSA, ElGamal, Elliptic curve cryptosystems, Public Key Infrastructure (PKI), Digital Signatures, Digital Certificates, Key Management and Key Distribution techniques	6	Knowledge of asymmetric encryption system
	Message Authentication and Integrity Verification: Message Authentication Codes, Modification Detection Codes, Hash Functions, Cryptographic Hash Functions, Merkle-Damgard Scheme, Preimage Resistance, Second Preimage Resistance, Collision Resistance, Random Oracle Model. Digital Signatures, RSA Digital Signature	8	Knowledge of message authentication and integrity verification
	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	Knowledge of entity authentication
	Key Management: Key Distribution Center (KDC), Symmetric key agreement protocol, Kerberos as a KDC, Certification Authorities for Public Keys, Role of Public Key Infrastructure	3	Concept about key management
	TOTAL	42	

Course Outcome	1. Basic knowledge of classical cryptosystems and the major goals of security 2. Private and public key cryptosystems with mathematical foundation 3. Key domain analysis of traditional and modern cryptosystems 4. Message authentication, digital signature, cryptographic hash functions, public key infrastructure 5. Entity authentication and key distribution infrastructures
-----------------------	---

Learning Resources	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. 4. Schneier, Bruce. Applied cryptography: protocols, algorithms, and source code in C. John Wiley & sons, 2007. 5. Katz, Jonathan, and Yehuda Lindell. Introduction to modern cryptography. CRC press, 2014.
---------------------------	--

Course Code	IT 5164N	Course Name	Medical Image Processing	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Image processing/signal processing	Co-requisite Courses	Algorithms, Information and Coding Theory	Progressive Courses	Medical Image Analysis, Machine and Deep Learning Medical Image Analysis and Healthcare Science, Machine and Deep Learning
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Introduction to medical imaging modalities and challenges in respective fields. Medical imaging challenges and reconstruction, enhancement and segmentation. Feature extraction, representations and selection on disease detection and classification. Complete CAD system design on different medical disease diagnosis
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Different Medical Imaging Methods Introduction to Imaging Methods in Medicine and Diagnosis. Introduction to basic physical and technological aspects of medical imaging. Different medical imaging modalities-Magnetic resonance imaging (MRI), x-ray imaging, CT, USG and microscopic imaging-principles of operation, relative merits and demerits	4	Introduction to image, medical image challenges and different medical imaging modalities
2	Compressed Sensing and MR image Reconstruction Introduction to Compressed sensing-theory and algorithms, signal sparsity, incoherence, different reconstruction algorithms, MR image reconstruction using CS/pMRI-theory, merits and demerits of different methods, MR Image Segmentation	6	Inverse problem: sub-sample image reconstruction, MRI and pMRI, ML/ DL methods in MRI/pMRI reconstruction
3	Medical Image Enhancement Pre-processing for medical images, de-noising techniques, filtering techniques for medical image enhancement	6	Noise cleaning and visual quality/ feature enhancement in medical images
4	Segmentation medical images Segmentation of Medical Images, information theoretic approaches, graph cut methods, SVM methods	6	Various medical Image segmentation techniques, object detection, RoI identification, ML/DL in object-background partitioning
5	Feature Extraction, selection and Classifier Design	4	Feature extraction, feature space

	Features in Biomedical Images, Feature extraction, Review of Selected Features, Feature matching. Classifier design introduction to machine learning		representations, scope of ML-DL in feature extraction
6	3D Image Analysis Feature extraction and analysis of 3D MR images	4	3D image reconstruction, Feature extraction, 2D CNN
7	CAD for Medical images Case studies: Diabetic Retinopathy Diabetic retinopathy, vascular net/vessel detection, Lesion detection on retinal images, stage detection of PDR based on classifier design Epilepsy on MR images Computer-aided Diagnosis of Epilepsy based on MR Images. Image Registration, Template Matching, Feature Extraction, Classification of Epileptic lesions CAD for Skin Biopsy Images Computer-aided Diagnosis of Skin Biopsy Images. Image Segmentation and Skin layer detection, Nuclei Detection, Diagnosis based on various image features X-Ray image analysis: Fracture Detection	12	Automated systems for Medical Image Analysis A complete system design from image acquisition to diagnosis, stage detection on different medical diseases
	TOTAL	42	

Course Outcome	By the end of the course, the students will be able to: CO1: Familiarization of different medical imaging techniques, advantages and challenges, multimodal imaging on medical domain CO2: Medical Image acquisition challenges, MRI and pMRI image reconstruction CO3: Medical image enhancement and object/Roi detection schemes CO4: Feature extraction, representation and selection for medical image analysis CO5: Complete CAD system design on different/multi-modal imaging on different medical image problems
-----------------------	--

Learning Resources	1. Gonzalez R. C. y Woods, R. E. "Digital Image Processing" Prentice Hall 3rd Ed. (2007) 2. Davies, E.R. "Computer and Machine Vision: Theory, Algorithms, Practicalities" Academic Press 4th Ed. (2012), 3. Bushberg, J. T., Seibert, J. A., Leidholt, E. M. and Boone, J. M. "The essential physics of medical imaging" Wolters Kluwer and Lippincott Williams & Wilkins 3rd Ed. (2012) 4. Frackowiack et al "Human Brain Function" Academic Press 2nd Ed. (2004), 1144 pgs.
---------------------------	---

Course Code	IT 5165N	Course Name	Embedded Systems and HW-SW Co-Design	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic and Circuit Design, Microprocessors, microcontrollers, Computer Organization and Architecture, Programming concepts.	Co-requisite Courses	Real Time Systems, Cyber Physical Systems	Progressive Courses	Cyber Physical Systems and Security
Course Offering Department		Information Technology		Data Book / Codes/Standards	NA

Course Objectives	By the end of this course, students will: Gain an in-depth understanding of embedded system architectures, design challenges, and applications. Learn systematic approaches to hardware–software co-design, partitioning, and integration. Develop skills in modeling, simulation, verification, and implementation of embedded solutions. Apply co-design principles to optimize performance, cost, and power efficiency. Engage with state-of-the-art tools, platforms, and standards used in industry
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	M1. Introduction to Embedded Systems Definitions, characteristics, and design metrics (performance, cost, power); Embedded vs. general-purpose computing; Application domains; Overview of development lifecycle	6	MO1: Explain key concepts and constraints of embedded systems and their design space.
2	M2. Embedded System Architecture Processor architectures (CISC, RISC, ARM, DSP); Memory hierarchy and management; I/O subsystems; Buses and interconnects; Real-time constraints	6	MO2: Analyze embedded architectures and their suitability for given applications.
3	M3. Hardware–Software Co-Design Fundamentals Co-design methodology; Design space exploration; Hardware–software partitioning; Interface design	6	MO3: Apply co-design principles to map functionality between hardware and software.
4	M4. Modeling and Simulation System-level modeling (UML, SysML); Transaction-Level Modeling (TLM); Co-simulation approaches; Tools such as MATLAB/Simulink, SystemC	6	MO4: Develop system-level models and evaluate functional correctness.
5	M5. Real-Time Systems and OS Support Real-time operating systems (RTOS) concepts; Scheduling algorithms; Inter-task communication; Device drivers; Case study with FreeRTOS or Zephyr	6	MO5: Implement RTOS-based scheduling and communication for embedded applications.

6	M6. Implementation Platforms FPGA and SoC platforms; High-Level Synthesis (HLS); Toolchains for ARM, Xilinx, Intel; Cross-compilation; Bootloaders	6	MO6: Use FPGA/microcontroller toolchains to synthesize and program embedded designs.
7	M7. Verification, Testing, and Optimization Functional verification; Timing analysis; Power optimization; Design-for-testability (DFT); Hardware debugging techniques	3	MO7: Apply verification techniques and optimize embedded designs.
8	M8. Emerging Trends and Case Studies Cyber-Physical Systems, AI-accelerated embedded platforms, Edge computing, Safety-critical systems, IoT gateways; Industry case studies	3	MO8: Evaluate cutting-edge applications and identify research challenges.
	TOTAL	42	

Course Outcome	After successful completion, students will be able to: CO1: Analyze requirements and constraints for embedded system applications. CO2: Apply hardware–software partitioning and mapping techniques for system design. CO3: Design embedded architectures integrating microprocessors/microcontrollers, peripherals, and communication interfaces. CO4: Use modeling and simulation tools to evaluate design trade-offs. CO5: Implement and validate embedded systems on target platforms (e.g., FPGA, SoC, or microcontroller boards). CO6: Critically evaluate emerging trends and research challenges in embedded system co-design.
-----------------------	---

Learning Resources	Textbooks 1. Marilyn Wolf, <i>Computers as Components: Principles of Embedded Computing System Design</i>, 4th ed., Morgan Kaufmann, 2016. 2. Frank Vahid and Tony Givargis, <i>Embedded System Design: A Unified Hardware/Software Introduction</i>, Wiley, 2nd ed., 2017. 3. Giovanni De Micheli and Luca Benini, <i>Networks on Chips: Technology and Tools</i>, Morgan Kaufmann, 2006. Reference Books 1. Peter Marwedel, <i>Embedded System Design</i>, 3rd ed., Springer, 2021. 2. Shuvra S. Bhattacharyya et al., <i>Hardware/Software Co-Design of Embedded Systems</i>, Springer, 2010. 3. Wayne Wolf, <i>High-Performance Embedded Computing: Architectures, Applications, and Methodologies</i>, Morgan Kaufmann, 2007. 4. Andrew N. Sloss, Dominic Symes, Chris Wright, <i>ARM System Developer's Guide</i>, Morgan Kaufmann, 2004. Research & Industry Standards • IEEE Transactions on Embedded Computing Systems (TECS) • ACM Transactions on Design Automation of Electronic Systems (TODAES) • ARM Architecture Reference Manuals • Xilinx & Intel FPGA documentation • FreeRTOS and Zephyr RTOS official docs
---------------------------	--

Course Code	IT 5166N	Course Name	Advance Data Science	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Programming Concepts, Mathematics for Data Science: Linear Algebra, Probability, Statistics, Basic Machine Learning Concepts, Database Management, Algorithms & Data Structures	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Overview of Python and Data Science Python Data Types and Structures, Control Flow and Functions in Python, Object-Oriented Programming (OOP) in Python, Working with Files and I/O Operations	4	Gain proficiency in basic Python programming tailored to data science.
2	NumPy: Arrays, Broadcasting, Vectorized Operations Pandas: DataFrames, Series, Data Cleaning, Handling Missing Data Matplotlib & Seaborn: Data Visualization, Plot Customization Data Wrangling and Transformation	4	Perform data manipulation, cleaning, and visualization using Python tools.
3	Descriptive Statistics and Data Distributions Handling Outliers and Missing Values Feature Engineering and Feature Scaling Data Imbalance and Sampling Techniques Correlation Analysis and Hypothesis Testing	3	Conduct statistical analysis and prepare features for modeling.
4	Scikit-Learn Basics: Pipeline, Model Selection, Metrics Supervised Learning (Linear Regression, Decision Trees, SVM, k-NN) Unsupervised Learning (K-Means, PCA, DBSCAN, Hierarchical Clustering) Model Evaluation: Bias-Variance Tradeoff, Cross-Validation, Grid Search Ensemble Learning (Random Forest, Gradient Boosting, XGBoost)	4	Build, evaluate, and optimize machine learning models using Scikit-Learn.
5	Introduction to Neural Networks: Perceptron, Backpropagation Deep Learning with TensorFlow and Keras Convolutional Neural Networks (CNNs) for Image Processing Recurrent Neural Networks (RNNs) and LSTMs for Sequential Data Transformers and Attention Mechanisms Model Optimization and Regularization	5	Implement deep learning models for structured and unstructured data.

6	Text Preprocessing with NLTK and spaCy Word Embeddings (Word2Vec, GloVe, BERT) Sentiment Analysis and Named Entity Recognition (NER) Text Classification and Sequence Modeling	3	Apply NLP techniques for extracting insights from text data.
7	Introduction to Big Data and Hadoop Ecosystem Data Processing with PySpark Streaming Data Analysis with Kafka & Spark Streaming Cloud-based Machine Learning (Google Cloud, AWS, Azure)	3	Use distributed tools to handle and analyze large-scale data streams.
8	Advanced Data Visualization using Plotly and Dash Dashboarding with Tableau and Power BI Case Studies in Finance, Healthcare, and E-Commerce	2	Create interactive visualizations and BI dashboards for insights.
9	Model Deployment using Flask, FastAPI ML Model Serving with Docker & Kubernetes ContinuousIntegration & Deployment (CI/CD) for ML Pipelines Monitoring and Maintaining Machine Learning Models	2	Deploy machine learning models using industry-standard practices.
10	Real-world Data Science Project Implementation Research Paper Writing & Presentation Ethics and Bias in AI & Data Science	4	Implement end-to-end projects with ethical and research considerations.
11	Overview of Python and Data Science Python Data Types and Structures Control Flow and Functions in Python Object-Oriented Programming (OOP) in Python Working with Files and I/O Operations	2	Gain proficiency in basic Python programming tailored to data science.
12	NumPy: Arrays, Broadcasting, Vectorized Operations Pandas: DataFrames, Series, Data Cleaning, Handling Missing Data Matplotlib & Seaborn: Data Visualization, Plot Customization Data Wrangling and Transformation	2	Perform data manipulation, cleaning, and visualization using Python tools.
13	Descriptive Statistics and Data Distributions Handling Outliers and Missing Values Feature Engineering and Feature Scaling Data Imbalance and Sampling Techniques Correlation Analysis and Hypothesis Testing	2	Conduct statistical analysis and prepare features for modeling.
14	Scikit-Learn Basics: Pipeline, Model Selection, Metrics, Supervised Learning (Linear Regression, Decision Trees, SVM, k-NN) Unsupervised Learning (K-Means, PCA, DBSCAN, Hierarchical Clustering) Model Evaluation: Bias-Variance Tradeoff, Cross-Validation, Grid Search Ensemble Learning (Random Forest, Gradient Boosting, XGBoost)	4	Build, evaluate, and optimize machine learning models using Scikit-Learn.
	Total	44	

Course Outcome	Upon successful completion of this course, students will: • Gain proficiency in data science, machine learning, and deep learning. • Learn how to preprocess and visualize data effectively. • Develop expertise in building machine learning models using Python libraries. • Understand deep learning frameworks and their applications. • Implement big data analytics with Python-based tools. • Deploy machine learning models in real-world scenarios.
-----------------------	--

Learning Resources	1. Python Data Science Handbook – Jake VanderPlas 2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron 3. Deep Learning with Python – François Chollet 4. Pattern Recognition and Machine Learning – Christopher M. Bishop 5. Data Science from Scratch – Joel Grus 6. Mining of Massive Datasets – Jure Leskovec, Anand Rajaraman, Jeffrey Ullman 7. Introduction to Machine Learning with Python – Andreas C. Müller, Sarah Guido 8. Python for Data Analysis – Wes McKinney
---------------------------	---

2nd Semester Courses Syllabi

M. Tech in Information Technology

Course Code	IT 5201N	Course Name	Advanced Database Management Systems	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Programming and Data structure, Discrete Mathematics	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	To learn how to design and manage databases Covers different DBMS concepts such as, Transaction Processing, Concurrency Control and Recovery database Understand database security and authorization Provide the idea of distributed database and data mining
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: Database, Database Management Systems, Structures of Relational Databases, Integrity Constraints.	2	Knowledge of relational database
2	Relational Database Design: Functional Dependency, Normal Forms, BCNF, 4NF, 5NF, Inclusion Dependency and Template Dependency.	4	Understand the database design and normalization
3	Storage Strategies: File organization, Indexing, Hashing Techniques, B tree and B+ tree, multiple key access.	5	Familiar with database storage structures and access techniques
4	Query Processing and Optimization: Evaluation of Relational Algebra Expressions, Join strategies, Query Optimization Algorithms.	4	Understand the techniques for query optimization
5	Transaction processing: Transaction concept, Serializability, Testing for serializability, Lock Based Protocols, Deadlock, Timestamp Based Protocols, Validation Based Protocols, Multiple Granularity. Recovery Techniques, Remote Backup Systems.	4	Familiar with the basic issues of transaction processing and

			concurrency control
6	Database Security: Levels of database security, discretionary access control, mandatory access control, Introduction to statistical database security.	4	Knowledge of database security
7	Distributed Databases: DDBMS architectures, Data Fragmentation, Replication and Allocation Techniques, Distributed query processing, Distributed transactions and concurrency management, deadlock management.	7	Understand distributed Database and Distributed transactions
8	Object Oriented Database: Object, object identity, object reference, architecture of Object-oriented and Object Relational database.	4	Idea of Object Relational database.
9	Data Warehousing and Data Mining: Introduction to Data warehouse and OLAP, Data Warehouse architecture, Introduction to Data mining, Classification of data mining techniques.	4	Understanding data warehouse and data mining techniques
10	Emerging Database Models: Multimedia database, web database, Mobile database, GIS, Gnome databases.	4	Learn multimedia, mobile database
	TOTAL	42	

Course Outcome	The following are the major outcome of this course: I. To design, construct and maintain a database and various database objects using procedural language constructs. II. To provide detailed knowledge of Transaction, concurrency and recovery, security strategies of DBMS. III. Describe the design of Distributed Databases and the basic concept of Data warehousing and Data mining. IV. To provide future research techniques for implementing single site DBMS.
-----------------------	--

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. G. Pelagatti and S. Cerri, "Distributed Databases: Principles and Systems", Mc Graw Hill, 2008
---------------------------	---

Course Code	IT 5202N	Course Name	Network and Information Security	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Number Theory, Discrete Mathematics, Computer Networking	Co-requisite Courses	Cryptography applications	Progressive Courses	Design of secure networks
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	This course aims towards presenting an overview of security requirements and challenges in network communication. This course will include the concepts of internetworking and various goals of security at different layers of the ISO/OSI model. It focusses on addressing those network security challenges using cryptographic techniques and protocols. This also includes state-of-the-art applications and practices that are implemented to provide email and web security.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Networking and Internetworking Concepts The ISO Open Systems Interconnection (OSI) model of communication, Technology underlying the TCP/IP Internet protocol suite and the architecture of an internet	3	Basics of Networks
2	Classical Cryptography and Overview: Model of secure network communication, Security goals and services, Overview of attacks, Security Architecture for Open System Interconnection (OSI), cryptanalysis	4	Introductory view of cryptographic concepts
3	Cryptographic Techniques: Introduction to Substitution Techniques, Transposition Techniques, Encryption and decryption, Symmetric and Asymmetric Key Cryptography, Key Range and Key Size	4	Different types encryption techniques
4	Private Key Cryptography: Symmetric Encryption. Definitions. Chosen-Plaintext Attack. Chosen-Ciphertext Attack, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, Variations on DES - RC4	8	Detailed understanding of the private key cryptography and the existing techniques in practice
5	Public Key Cryptography: RSA, ElGamal, DSA, Elliptic curve cryptosystems, Public Key Cryptography standard (PKCS), PKI, Digital Certificates, and Key management techniques	6	Understanding the different public key cryptography techniques and their applicability
6	Message Integrity and Message Authentication:	4	Message integrity and

	Message integrity, Random Oracle Model, Message authentication, Cryptographic Hash Functions, Digital Signature, Blind Signature		authentication algorithms in the view of computer networks
7	Security Protocols at Various Layers: E-mail System, Pretty good privacy (PGP), Secure/Multipurpose Internet Mail extension. Security services at transport layer, SSL/TLS architecture, four protocols, SSL Message format. Security at the network layer, IP Security (IPSec), Modes of IPSec, Security Protocols, Security Association, Security Policy, Internet Key Exchange (IKE)	8	Understanding the application layer protocols
8	System Security: Buffer overflow and malicious software, malicious programs, intrusion detection system. Firewalls: Definition, Construction, Working principles	2	Understanding the aspects of system security
9	Media Security: Data hiding, Steganography & Steganalysis, Digital watermarking, Visual Cryptography	3	The aspects of media security and relevant algorithms
	TOTAL	42	

Course Outcome	1. Identify network security threats at different layers of the ISO/OSI model and propose solutions to overcome those. 2. Achieve secure network communications 3. Verify the integrity of information exchanges in computer networks 4. Develop email and file transfer security systems 5. Develop SSL or Firewall based solutions against security threats
-----------------------	---

Learning Resources	Essential Reading: 1. Behrouz A. Forouzan and D. Mukhopadhyay - Cryptography & Network Security, McGraw Hill. 2. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall. 3. Douglas E. Comer and David L. Stevens. Internetworking with TCP/IP: Volume II: Design, Implementation, and Internals. Prentice Hall. References: 1. Douglas R. Stinson, Cryptography: Theory and Practice, Chapman and Hall.
---------------------------	--

Course Code	IT 5203N	Course Name	Internet and Distributed Computing	Course Category	PC	L	T	P
						3	0	0

Pre-requisite Courses	Computer Networks	Co-requisite Courses	Advanced Database Management System	Progressive Courses	M.Tech Projects
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	1. To learn basics of distributed computing and the power of collective effort in computing 2. To understand Internet as a distributed system 3. To understand the use of distributed computing techniques in real-life applications
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Review of TCP/IP Protocol Stack: IPv4, IPv6, TCP, UDP, ARP, ICMP, SMTP etc.	4	Understanding of Computer Networks
2	Introduction to Distributed Systems, Internet as a Distributed System, Transparency and Openness in Internet, RFCs	5	Understanding of Philosophy of Internet
3	Model of Distributed Computation; Synchronous, Asynchronous Networks, and Anonymous Networks; Introduction to Design and Analysis of Concurrent Algorithms, Message Complexity	6	Stepping into Distributed Systems
4	World Wide Web as a Distributed Document Based System, Client-Server Architecture in Web, Document Model: Markup languages,	5	Understanding Internet as a Distributed System
5	Election algorithms, Impossibility results, Lower bound for synchronous networks	6	Learning Elementary Algorithms
6	Logical clock; Synchronization; Mutual exclusion algorithms	4	Understanding the Role of Time in Computing
7	Distributed Consensus and Fault tolerance, Fault tolerance in Internet	4	Understanding Faults and their Management
8	Internet bot, Web crawler and Search engines	4	Understanding Internet as a Distributed System
9	IoT evolution, Basics of IoT, Applications of IoT in home automation, healthcare, agriculture, and industrial applications, Emerging directions in IoT.	4	Understanding the use of Distributed

			Computing in Real Life
	TOTAL	42	

Course Outcome	1. Learning of basics of distributed computing and the power of collective effort in computing 2. Understanding of Internet as a distributed system 3. Understanding of the use of distributed computing techniques in real-life applications
-----------------------	---

Learning Resources	1. Ajoy D. Kshemkalyani and MukeshSinghal. Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press 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
---------------------------	---

Course Code	IT 5221N	Course Name	Multimedia Coding and Compression	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Basic knowledge of multimedia systems and information and coding theory	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to image and video Compression , Video signal representation, Lossless Coding Techniques, Entropy Coding, Huffman and Arithmetic Coding, Predictive Coding.	2	Basic about compression
2	Transform Coding Transforms, block-wise transform coding, orthogonal and orthonormal transform, Transform coding gain, Bit allocation for transform coefficients, DCT & DFT, Threshold coding, Typical coding artifacts, Fast implementation of the DCT, Haar Transform, KarhunenLoève Transform (KLT),Walsh-Hadamard Transform (WHT)	7	Knowledge about transform coding
3	Wavelet Coding Embedded Zero Tree Wavelet (EZW), Set-Partitioning in Hierarchical Trees (SPIHT), Embedded Block Coding with Optimized Truncation (EBCOT), Wavelet Difference Reduction (WDR), Space – Frequency Quantization (SFQ), Stack- Run (SR), Geometric Wavelet (GW), Filters and Filterbanks	7	Knowledge about wavelet coding
4	Video Coding Standards H.261/H.263, MPEG2/MPEG4, H.264	5	Knowledge of different video coding standards
5	Error-resilient coding techniques Error propagation in video coding, Spatial error resilience coding, Temporal error resilience coding	4	Different error resilient coding techniques
6	Image Compression JPEG, JPEG 2000, Embedded Block Coding with Optimized Truncation (EBCOT) in JPEG 2000. JPEG XT	5	JPEG image Compression technique
7	Audio Compression PCM, ADPCM in Speech Coding, Linear Predictive Coding, MP3, MPEG	4	Knowledge of audio compression technique
8	Video Compression, Motion Estimation	8	Knowledge of video

	Motion Estimation Techniques, Temporal correlation and motion compensation, Differential methods, Block matching, Sub-pixel accuracy, Fast algorithm, Rate-constraint Motion Estimation Standard Video Compression Techniques MPEG, MPEG2/MPEG4, H.261/H.263, H.264		compression techniques
9	TOTAL	42	

Course Outcome	
-----------------------	--

Learning Resources	Text Books: 1. Sun, Huifang, and Yun Q. Shi. Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards. CRC press, 2008. 2. Gibson, J. D., Berger, T., Lookabaugh, T., Baker, R., & Lindbergh, D. (1998). Digital compression for multimedia: principles and standards. 3. Steinmetz, R., & Nahrstedt, K. (2002). Multimedia fundamentals, volume 1: media coding and content processing. Pearson Education. Reference Books: 1. Steinmetz, Ralf, and Klara Nahrstedt. Multimedia: Computing, Communications and Applications: Media Coding and Content Processing. Prentice Hall PTR, 2002. 2. Nelson, Mark, and Jean-Loup Gailly. The data compression book. New York: M & t Books, 3. Richardson, Iain E. H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons, 2004.
---------------------------	---

Course Code	IT 5222N	Course Name	Mobile Computing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Networking, Operating System and basic communication .	Co-requisite Courses	Wireless Communication, Computer Networks Laboratory, Information Security / Network Security.	Progressive Courses	Advanced Wireless Networks, Internet of Things (IoT), Edge and Cloud Computing, Ad-Hoc and Sensor Networks, Mobile Application Development.
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Introduce the basic concepts of mobile communication and computing. Familiarize students with different wireless technologies and standards including GSM, CDMA, 3G, 4G. Understand mobile IP, mobility adaptation, and energy-efficient computing. Explain mobile computing architectures and security aspects. Analyze protocols, routing techniques, and performance issues in mobile and ad-hoc networks.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to mobile computing	3	Understand the fundamentals and importance of mobile computing in today's technology landscape.
2	Wireless and Cellular network, channel allocation, multiple access	3	Explain the structure and functioning of wireless communication networks and resource allocation techniques.
3	1G, 2G, systems, GSM standards, architecture	3	Identify features and architecture of early mobile systems and GSM.
4	Location management, Handoffs, Authentication	3	Describe location tracking, handoff strategies, and authentication mechanisms.
5	2G CDMA, 3G CDMA, 4G standards and advances	4	Analyze CDMA-based systems and evolution to 3G and 4G technologies.

6	IEEE 802.11 WLAN	4	Understand WLAN architecture, protocols and standards.
7	Bluetooth, HiperLAN architecture, comparison of wireless technologies	4	Compare different short-range wireless communication technologies.
8	Mobility adaptation, process migration, mobile IP	4	Understand challenges and solutions for mobility and process migration.
9	Mobile Ad-hoc networking. MAC protocols, Routing	4	Analyze the functioning and routing protocols in mobile ad-hoc networks.
10	Energy efficient computing, Impact of mobility on algorithms	4	Explore energy optimization strategies in mobile systems and their algorithmic impacts.
11	Mobile Computing Architecture	2	Describe architectural models for mobile computing.
12	Mobile Computing through Telephony	2	Explain telephony-based mobile computing concepts.
13	Security Issues in Mobile Computing	2	Discuss security concerns and solutions in mobile computing environments.
	Total	42	

Course Outcome	By the end of the course, students will be able to: CO1: Explain the key concepts of mobile communication and mobile computing. CO2: Analyze and compare various generations of wireless technologies including GSM, CDMA, 3G, 4G. CO3: Evaluate protocols used for mobility, routing, and energy-efficient computing. CO4: Design and assess solutions for mobile and wireless networks with security considerations. CO5: Apply knowledge of mobile architecture and adapt existing systems for mobility.
-----------------------	--

Learning Resources	1. Fundamentals of Mobile Computing by Pattnaik Mall, PHI. 2. Mobile Computing, by TalukderAsoke K. Hasan Ahmed and Roopa Yavagal, 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	IT 5223N	Course Name	Approximation Algorithms	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Design and Analysis of Algorithm	Co-requisite Courses		Progressive Courses	
Course Offering Department	Information Technology		Data Book / Codes/Standards	None	

Course Objectives	Design approximation algorithms to find provably near-optimal solutions analyse the complexity and approximability of important optimisation problems Learn various approximation algorithm design techniques ranging from combinatorial techniques to sophisticated LP based and probabilistic methods Understand the theoretical limits of approximation
-------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction: a. Preliminaries and Basic Definitions b. Absolute Performance Guarantees c. Relative Performance Guarantees	4	Understand the Preliminaries
2	Combinatorial Algorithms: Approximation Schemes: a. Approximation Scheme for Scheduling b. Approximation Scheme for Knapsack c. Fully Polynomial Approximation Schemes d. Pseudo-Polynomial Algorithms e. Strong N P-completeness and FPAS	6	Understand the Approximation Schemes
3	Vertex Cover and Set Cover a. Approximating Vertex Cover b. Approximating Weighted Vertex Cover i) A Randomized Approximation Algorithm ii) The Nemhauser Trotter Algorithm iii) Clarkson's Algorithm b. Improved Vertex Cover Approximations i) The Nemhauser-Trotter Algorithm Revisited ii) A Local Ratio Theorem iii) An Algorithm for Graphs Without Small Odd Cycles iv) The Overall Algorithm c. Approximating Set over	6	Know approximation for vertex cover problem
4	Knapsack a) A pseudo-polynomial time algorithm for knapsack b) An FPTAS for knapsack c) Strong NP-hardness and the existence of FPTAS's	4	Know approximation for Knapsack problem
5	Bin Packing a) Asymptotic Approximation Scheme i) Restricted Bin Packing ii) Eliminating Small Items iii) Linear Grouping iv) APAS for Bin Packing b) Asymptotic Fully Polynomial Scheme i) Fractional Bin Packing and Rounding ii) AFPAS for Bin Packing iii) Near-Absolute Approximation	6	Understand the Approximation for Bin Packing problem

6	LP-Based Algorithms Introduction to LP-Duality a. The LP-duality theorem b. Min–max relations and LP-duality c. Two fundamental algorithm design techniques	4	Understand LP duality principles to analyze approximation
7	Set Cover via Dual Fitting a) Dual-fitting-based analysis for the greedy set cover algorithm b) Generalizations of set cover	4	Understand dual-fitting technique to analyze the performance of approximation Set Cover problem
8	Maximum Satisfiability a) Dealing with large clauses b) Derandomizing via the method of conditional expectation c) Dealing with small clauses via LP-rounding d) A 3/4 factor algorithm	4	Approximation algorithms for the Maximum Satisfiability problem
9	Facility Location a) An intuitive understanding of the dual b) Relaxing primal complementary slackness conditions c) Primal–dual schema based algorithm d) Analysis	4	Approximation algorithms for the Facility Location problem
	TOTAL	42	

Course Outcome	Ability to identify the design technique used in a given approximation algorithm Design and analyze approximation algorithms using various combinatorial approximation techniques To apply more sophisticated techniques like randomization and LP based optimization to computationally hard problems
-----------------------	--

Learning Resources	1. Vazirani, Vijay V. Approximation algorithms. Springer Science & Business Media, 2013. 2. Motwani, Rajeev. "Lecture notes on approximation algorithms: Volume I." Dept. Comput. Sci., Stanford Univ., Stanford, CA, Tech. Rep. CS-TR-92-1435 (1992).
---------------------------	---

Course Code	IT 5224N	Course Name	Real Time Systems	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Introductory course on Embedded Systems, First course on Operating Systems	Co-requisite Courses	Advanced Computer Architecture	Progressive Courses	Embedded system
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	- Understand the design and analysis of real-time operating systems (RTOS) and scheduling algorithms. - Explore timing constraints, task prioritization, and synchronization in real-time environments. - Study hardware and software architectures used in real-time applications. - Provide practical exposure to developing and analyzing real-time systems in embedded and control domains.
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Real Time Systems Introduction to Real time Embedded System, need for a real-time system, different kinds, Embedded system Design cycle, Types of Real Time systems, Real Time Applications and features, Issues in real time computing, aspects of real-time systems, Performance measures of Real Time System, real-time requirement specifications, modelling/verifying design tools (real time UML, state charts, etc.,).	4	Understand the basic concepts, classifications, and characteristics of real-time systems and their role in time-critical applications.
2	Hardware for Real Time Systems Selection criteria for Real time system - Hardware and Software perspective, need for partitioning, criteria for partitioning, System Considerations, Basic development environment-host vs target concept, CPU features – Architecture, on-chip peripherals, Real time implementation considerations, pipeline, bus architecture, Fast Interrupt Response Manager, Introduction to Interrupts, Interrupt vector table, interrupt programming, Pipeline and Parallelism concepts.	8	Understand the architecture and components used in designing real-time systems.
3	On-chip peripherals and Communication protocols Role of peripherals for Real time systems, On-Chip peripherals & hardware accelerators, Peripherals [Direct Memory Access, Timers, Analog to Digital Conversion (ADC), DAC, Comparator, Pulse Width Modulation (PWM)], Need of real time Communication, Communication Requirements, Timeliness, Dependability, Design Issues, Overview of Real time communication, Real time Communication Peripherals – I2C, SPI & UART.	8	Understand interface and utilize on-chip peripherals and implement communication protocols essential for real-time embedded system operations.

4	Embedded Software and RTOS Software Architecture of real time System, Introduction to RTOS, role of RTOS, Real time kernel, qualities of good RTOS, Functionalities of RTOS – Task Management, I/O management, Memory management, Inter Task Communication, Task, Task states, Task control block, attributes of TCB, Context switching, Interrupts handling, Multiprocessing and multitasking.	6	Understand to develop and manage embedded software using real-time operating system (RTOS)
5	Scheduling, Synchronization and Inter task communication in RTS Basic Concepts for Real-Time Task Scheduling, Scheduling: definitions, Overview of Scheduling policies, Task Synchronization – Need of synchronization, shared data problems and its ways of handling, Role of Semaphore, types of semaphores, Inter task communication – Need of communication, Message Mailbox and Message Queues, RTOS problems - Priority inversion phenomenon, Deadlock phenomenon and steps to handle them.	8	Understand timing constraints, task models, and priority-based scheduling techniques and skill to integrate resource access mechanisms.
6	RTOS Programming I: MicroC/OS-II and VxWorks Basic functions and types of RTOSes, RTOS mCOS – II, RTOS VxWorks	4	Understanding RTOS programming for real time application development in this platform
7	RTOS Programming II: Windows CE, and RT Linux Windows CE, RTLinux.	4	Understanding RTOS programming for real time application development in this platform
	TOTAL	42	

Course Outcome	● Ability to design and develop a real time system from the given specifications and requirements. ● Ability to understand timing constraints, task models, and priority-based scheduling techniques and skill to integrate resource access mechanisms. ● Evaluate the performance and correctness of real-time systems using analytical tools. ● Understand the architecture and functionalities of real-time operating systems. ● Develop simple real-time applications using RTOS or embedded development platforms.
-----------------------	---

Learning Resources	Text Books: 1. Real-Time Systems by Jane W. S. Liu Prentice Hall; 1 edition ISBN: 978-0130996510 2. Krishna .C.M “Real Time Systems” Mc-Graw Hill Publication. 3. Hamid A. Toliyat and Steven G. Campbell, “DSP based Electromechanical Motion Control” CRC Press, 2003, ISBN 9780849319181. 4. Embedded systems architecture, programming and design (2e) - by Raj Kamal – TMH. Reference Books:
---------------------------	--

	1. Jean J Labrosse, "Embedded System Design blocks", CMP books, Second Edition, ISBN 0-87930-604-1. 2. Embedded Systems Architecture – by Noergaard, ELSEVIER
--	---

Course Code	IT 5225N	Course Name	Machine Learning	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Linear algebra: Simple linear algebra. Basic Programming concepts. Calculus: Some differential calculus.	Co-requisite Courses	Discrete Mathematics, Probability and Statistics	Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	• To learn fundamental Machine Learning techniques and principles • To learn mathematics behind Machine Learning SOA algorithms • To learn implementation challenges of SOA Machine Learning Techniques
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to Machine Learning Theory of Learning, Early AI vs. Machine Learning, Well-posed Learning Problem, Data Dimensionality, Curse of Dimensionality, Applications of Machine Learning (ML), Classes of ML Techniques	3	Introduction to ML
2	Mathematical Preliminaries Probability and statistics, concepts of linear algebra, distribution functions, statistical learning, Expectation and Moments	3	Mathematical base development
3	Concept Learning Learning from examples, Representing hypotheses, Inductive learning hypothesis, General to Specific Ordering, Version Space, Find-S algorithm, List-then-Eliminate method, Candidate Elimination Algorithm, Inductive leap, Inductive bias	3	Ordering of Hypothesis space and concept learning
4	Decision Trees Decision tree logic, Prediction using decision trees, Entropy, Information Gain, Decision tree using ID3, Restriction vs. Preferential Bias, Overfitting in Decision Trees, Decision Tree Pruning	3	Learning Decision Tree Classifier Principle
5	Bayesian Learning Bayes Theorem, MAP Hypothesis, ML Hypothesis, MAP Hypothesis Learner, Evolution of Posterior Probabilities, Bayes Optimal Classifier, Naive Bayes Classifier	3	Learning Bayes Classifier Principles
6	Instance based Learning Lazy Learners, concept of nearest neighbor, K-nearest neighbor, KNN classifier, Distance weighted KNN classifier	3	Learning KNN Classifier Principles

7	Linear Regression Classification vs. Regression, Univariate Linear regression, Multivariate Linear Regression, Least Squares Regression, LMS Weight Update Rule, Classification as a linear regression problem	3	Introduction to Regression and Linear Regression
8	Logistic Regression Linear vs. logistic regression, Sigmoid function, Loss function, Binary Cross Entropy, Linear separability, Multinomial logistic regression, One-hot vector, Categorical distribution, Generalized Cross Entropy, Binary Classification, Support Vectors and Margins, SVM Classification	4	Logistic Regression and SVM Classifier
9	Ensemble Techniques Ensemble learning, Multifold cross validation, Boosting, AdaBoost, Bagging, Subbagging, Random Forest	4	Ensembling multiple classifiers
10	Unsupervised Learning Unsupervised learning concepts, distance measures, concept of nearest neighbors in unsupervised learning, K-means clustering, Elbow method, Silhouette method, Noise handling	6	Learning Clustering Techniques and Principles
11	Feature Engineering Feature Selection, Wrapper vs. Filter Methods, Forward vs. Backward Feature Selection, Univariate and Multivariate Feature Selection, Dimensionality Reduction, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)	6	Feature Selection
	TOTAL	41	

Course Outcome	Comprehensive understanding of SOA Machine Learning algorithms, mathematical basis, tool and techniques.
-----------------------	--

Learning Resources	Text books 1. Christopher M. Bishop. Pattern Recognition and Machine Learning (Springer) 2. David Barber, Bayesian Reasoning and Machine Learning (Cambridge University Press). 3. Tom Mitchell. Machine Learning (McGraw Hill) 4. Richard O. Duda, Peter E. Hart, David G. Stork. Pattern Classification (John Wiley & Sons) Reference texts • Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer, 2009 (freely available online) • Hal Daumé III, A Course in Machine Learning, 2015 (in preparation; most chapters freely available online) • Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012 • Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007. • Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
---------------------------	--

Course Code	IT 5226N	Course Name	Cloud and Services Computing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	The students required to know about the basic computer networks, data structures, database managements, operating systems	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	To learn the following 1. Concepts of cloud computing, models, applications 2. Advantages and disadvantages of clouds 3. Cloud storage, cloud networks 4. Different cloud-computing services and platforms 5. Concepts of virtualization 6. Cloud file systems and computing paradigms
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to cloud computing Introduction, characteristics of cloud computing, cloud model, cloud service, cloud applications	6	This module provides an introduction to cloud computing.
2	Cloud concepts and technologies Virtualisation, load balancing, scalability, deployment, replication, monitoring, software defined network, service level agreement, billing	7	This module introduces various technologies used in cloud computing to students.
3	Cloud services and data platforms Compute services, storage services, database services, application services, content delivery services, analytics services, deployment and management services, cloud data platforms (e.g., Databricks, Snowflake)	8	This module discusses different cloud services and data platforms.
4	Hadoop and MapReduce Hadoop MapReduce job execution, Hadoop scheduler, Hadoop cluster setup	7	This module discusses file systems and computing paradigms.
5	Cloud application design Introduction, reference architecture, design methodology, data storage approaches	8	This module discusses

			designing cloud applications.
6	Cloud application development and case studies	6	This module discusses case studies for guiding through cloud application development.
	TOTAL	42	

Course Outcome	After completion of the course, students will have an understanding of the following 1. Concepts of cloud computing, models, applications 2. Advantages and disadvantages of clouds 3. Cloud storage, cloud networks 4. Different cloud-computing services and platforms 5. Concepts of virtualization 6. Cloud file systems and computing paradigms
-----------------------	--

Learning Resources	1. Cloud Computing: A Hands-On Approach By Arshdeep Bahga, Vijay Madisetti, 2. Fox, Armando, et al. "Above the clouds: A berkeley view of cloud computing." Dept. Electrical Eng. And Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28.13 (2009): 2009. 3. Cloud Computing: Concepts, Technology & Architecture, Book by Richardo Puttini, Thomas Erl, and Zaigham Mahmood, Prentice Hall.
---------------------------	--

Course Code	IT 5227N	Course Name	High-Performance and Parallel Computing	Course Category	PSE	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic and Circuit Design, Computer Organization and Architecture, Operating Systems, Programming concepts	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards	NA	

Course Objectives	The course aims to: Develop a deep understanding of architectures, models, and tools for high-performance and parallel computing. Enable students to design, analyze, and optimize parallel algorithms for large-scale scientific and engineering problems. Provide hands-on skills in programming for multi-core, GPU, and distributed-memory systems. Prepare students to evaluate system performance and scalability in HPC environments.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Module 1: Introduction to High-Performance Computing Topics: Evolution of HPC and its role in scientific & engineering computing. Flynn's taxonomy (SISD, SIMD, MISD, MIMD). Parallelism levels: data, task, instruction, pipeline. HPC architectures overview: multicore CPUs, GPUs, clusters, supercomputers.	6	MO1.1: Classify parallel architectures and parallelism types. MO1.2: Explain the evolution and relevance of HPC in modern research and industry.
2	Module 2: Parallel Programming Models and Paradigms Topics: Shared-memory model (threads, OpenMP). Distributed-memory model (message passing, MPI). GPU computing (CUDA, OpenCL basics). Hybrid programming models.	6	MO2.1: Distinguish between shared, distributed, and hybrid models. MO2.2: Write basic parallel programs in OpenMP and MPI.
3	Module 3: Parallel Algorithm Design Topics: Principles of parallel algorithm design: decomposition, mapping, granularity. Parallel patterns: map, reduce, scan, pipeline, stencil. Load balancing and task scheduling.	6	MO3.1: Apply decomposition techniques to design parallel algorithms. MO3.2: Implement common parallel computation patterns.

4	Module 4: Performance Analysis and Optimization Topics: Performance metrics: speedup, efficiency, scalability, Amdahl's law, Gustafson's law. Profiling and benchmarking tools. Memory hierarchy, cache optimization, communication optimization.	6	MO4.1: Analyze performance bottlenecks using theoretical models and profiling. MO4.2: Optimize memory and communication for HPC applications.
5	Module 5: GPU and Accelerator-based Computing Topics: GPU architecture and programming model. CUDA programming: threads, blocks, grids, shared memory. Optimization for GPUs: memory coalescing, occupancy, kernel fusion.	6	MO5.1: Develop simple GPU programs in CUDA. MO5.2: Optimize kernels for performance and memory usage.
6	Module 6: Scalable and Distributed Systems in HPC Topics: Cluster computing, cloud-based HPC. Fault tolerance and checkpointing. HPC file systems and I/O optimization.	6	MO6.1: Explain architectures and software stacks for scalable HPC systems. MO6.2: Implement basic distributed computing tasks with fault tolerance.
7	Module 7: Case Studies & Emerging Trends Topics: HPC applications: climate modeling, genomics, AI/ML acceleration, computational fluid dynamics. Exascale computing challenges. Energy-efficient HPC, quantum acceleration.	6	MO7.1: Analyze real-world HPC use cases and performance trade-offs. MO7.2: Discuss future trends in HPC and parallel computing research.
	TOTAL	42	

Course Outcome	Upon successful completion, students will be able to: CO1: Explain key concepts of parallelism, concurrency, and scalability in modern HPC systems. CO2: Model and analyze the performance of parallel programs using established metrics. CO3: Design and implement parallel algorithms using shared-memory, distributed-memory, and GPU paradigms. CO4: Apply optimization techniques for high performance on heterogeneous systems. CO5: Critically evaluate HPC systems, tools, and benchmarks for research and industrial applications.
-----------------------	---

Learning Resources	Textbooks - Quinn, M. J. <i>Parallel Programming in C with MPI and OpenMP</i>. McGraw-Hill, 2004. - Grama, A., Gupta, A., Karypis, G., Kumar, V. <i>Introduction to Parallel Computing</i>. 2nd Ed., Addison-Wesley, 2003. - Kirk, D. B., Hwu, W. W. <i>Programming Massively Parallel Processors: A Hands-on Approach</i>. Morgan Kaufmann, 3rd Ed., 2016. Reference Books - Pacheco, P. S. <i>An Introduction to Parallel Programming</i>. Morgan Kaufmann, 2011.
---------------------------	---

- | | |
|--|--|
| | <ol style="list-style-type: none">5. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. <i>Parallel Programming in OpenMP</i>. Morgan Kaufmann, 2001.6. Rauber, T., Rünger, G. <i>Parallel Programming: For Multicore and Cluster Systems</i>. Springer, 2013. |
|--|--|

Online & Supplemental Resources

- Top500.org – HPC system rankings and benchmarks.
- NVIDIA CUDA Toolkit documentation.
- OpenMP and MPI official documentation.
- LLNL tutorials on parallel computing.

Course Code	IT 5261N	Course Name	CAD Algorithms for VLSI	Course Category	O E	L	T	P
						3	0	0

Pre-requisite Courses	Digital Logic Design, Data Structures and Algorithms, Introduction to VLSI Design, Discrete Mathematics, Computer Architecture	Co-requisite Courses	VLSI Physical Design, Digital System Design	Progressive Courses	Advanced VLSI Physical Design Automation, High-Level Synthesis (HLS)
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	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.	6	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.	4	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 routing, etc. Special considerations for analog and mixed-signal design.	8	learn key physical design automation steps with emphasis on analog and mixed-signal design challenges.

7	Testability issues: Fault modelling and simulation, test generation, design for testability, built-in self-test. Testing SoCs.	6	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.	6	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	IT 5262N	Course Name	Computational Topology	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Basics of set theory, Basic idea about algorithms	Co-requisite Courses	Discrete Mathematics	Progressive Courses	Image processing applications, data science applications using topology
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to point set topology: Open and Closed Sets, Neighborhoods, Definition of Topology, Basis for Topology, Continuous maps, Connectedness, Path-connectedness Compactness, Separation, Quotient Topology	4	Introductory ideas about topology
2	Graph: Connected components, Curves and Knots, Planar graphs	6	Fundamental ideas about the different geometric entities
3	Surfaces: Two-dimensional Manifolds, Searching a Triangulation, Self-intersections, Surface Simplification	4	Ideas about the surfaces and their mathematical treatise
4	Complexes: Simplicial Complexes, Convex Set Systems, Delaunay Complexes, Alpha Complexes	4	Detailed idea about the representation of such systems
5	Homology: Homology groups, Matrix reduction, Relative homology, Exact Sequences	4	Understanding homology
6	Duality: Cohomology, Poincaré Duality, Intersection Theory, Alexander Duality	4	Understanding the different aspects of duality
7	Persistence: Persistent homology, Efficient Implementation, Stability, Extended persistence, Spectral Sequence	4	The idea of persistence and usage in homology
8	Morse Function: Generic smooth functions, Transversality condition, Piecewise linear functions, Reeb graphs	4	Ideas about Morse function and Reeb graphs for structural understanding of geometric objects
9	Applications: Image Segmentation, Elevation, Gene Expression, Local Homology for Plant Root Architecture	4	Getting ideas about the application of topology in image processing
	TOTAL	41	

Course Outcome	
-----------------------	--

Learning Resources	1. P. S. Alexandroff. Elementary Concepts in Topology. translated by A. E. Farley, Dover, New York, 1961. 2. H. Edelsbrunner. Geometry and Topology for Mesh Generation. Cambridge Univ. Press, England, 2001. 3. P. J. Giblin. Graphs, Surfaces and Homology. 2nd edition, Chapman and Hall, London, 1977. 4. Y. Matsumoto. An Introduction to Morse Theory. Amer. Math. Soc., Providence, Rhode Island, 2002. 5. J. W. Milnor. Topology from the Differential Viewpoint. Princeton Univ. Press, New Jersey, 1965. 6. J. R. Munkres. Topology. A First Course. Prentice-Hall, Englewood Cliffs, New Jersey, 1975. 7. J. R. Munkres. Elements of Algebraic Topology. Perseus, Cambridge, Massachusetts, 1984.
---------------------------	---

Course Code	IT 5263N	Course Name	IoT Systems	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Basic knowledge of Embedded Systems, Wireless and Computer Networks, Operating Systems	Co-requisite Courses	Embedded Systems Laboratory, Wireless Communication, Cloud Computing Fundamentals, Data Structures and Algorithms	Progressive Courses	Cyber-Physical Systems, Edge and Fog Computing, IoT Security and Privacy, Smart Systems and Applications, AI for IoT.
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Introduce the fundamental principles, architecture, and evolution of Internet of Things systems. Familiarize students with core networking and communication protocols in IoT environments. Enable understanding of open-source IoT infrastructures and cloud/fog-based integrations. Provide foundational insight into embedded computing and device-cloud communication. Develop basic understanding of security, privacy, and cryptographic approaches in IoT. Explore real-time data streaming, analytics, and distributed frameworks in IoT ecosystems.
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to internet of things, emergence of IoT, smartness in IoT, Improving quality of life	4	Understand the motivation, applications, and impact of IoT in daily life and society.
2	Basic of Internet technologies, IoT architecture, resource management, data management, communication protocols	6	Explain the layered architecture of IoT and core communication mechanisms.
3	IoT standards, open source vs close source, open source web infrastructure, openIoT architecture, cloud convergence	6	Evaluate open-source IoT tools and infrastructure, and understand cloud integration.
4	Intelligent cloud collaboration framework, device-cloud collaboration, introduction to fog computing	4	Analyze frameworks for device-cloud and fog-level collaboration.

5	IP as the IoT network layer, optimization of IP, IP for smart objects	4	Understand IP-based communication tailored to low-power smart devices.
6	Application protocols for IoT, IoT transport layer, IoT application layer	4	Identify transport/application protocols specific to IoT (e.g., MQTT, CoAP).
7	Data and analytics for IoT: streaming process, scalability, robustness, framework for distributed data analysis	6	Describe scalable and distributed approaches to IoT data processing.
8	Introduction of embedded computing: embedded device programming language, virtualization on embedded boards, cloud assisted cyber-physical systems	4	Explain basics of embedded programming, virtualization, and cyber-physical integration.
9	IoT security and privacy, network and transport layer challenges, security framework: light weight cryptography	4	Recognize threats in IoT and apply lightweight security frameworks.
	TOTAL	42	

Course Outcome	After completion of the course, students will gather knowledge on: ● IoT architecture and smartness of IoT ● IoT open source web infrastructure ● IoT network and transport layer protocols ● IoT Data analytics ● Introductory overview on embedded computing and IoT systems ● IoT security
-----------------------	--

Learning Resources	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	IT 5264N	Course Name	DSP Algorithms	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Concept on signals and systems, digital systems design, Optimization-conventional and meta-heuristic	Co-requisite Courses	Medical image processing , Soft computing	Progressive Courses	Machine Learning, Deep Learning, Medical Image Analysis, Computer Vision, Image Processing
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Digital filter design, algorithmic issues, application specific filter design knowledge. Optimization in filter design making them suitable in different applications, hardware and software relaxation of filter
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Introduction to digital signal processing-its benefits and applications, Sampling and quantization -analog to digital conversion (ADC), Digital to analog conversion (DAC)	2	Relative merits and challenges of DSP over analog version
2	Mathematical operations on discrete-time signals, linear time invariant systems, causality, stability, difference equations, frequency response, discrete Fourier series, , Implementation of discrete-time systems	4	Introduction of discrete signals and systems, frequency description of discrete signals
3	Z-transform - definition, properties of Z-transform, system function, digital filter implementation from the system function, region of convergence in the Z plane, determining filter coefficients from the singularity locations, geometric evolution of Z transform in the Z plane, relationship between Fourier transform and Z transform, inverse Z transform.	4	Frequency domain role of Z-transform on discrete signals, pole-zero concept, stability of system
4	Digital filter structures: system describing equations, filter categories, direct form I and II structures, cascade and parallel communication of second order systems	4	Digital filter design, types and challenges
5	FIR filter design techniques: Fourier Transform Design, Windowing method, combining DFT and window method for designing FIR filter, frequency sampling method for designing FIR filter, Optimal Design method	6	FIR filter design, techniques, challenges. Optimal design
6	IIR filter design techniques: Approximation theory, Impulse invariant and bilinear transformations, Frequency transformations	6	IIR filter design, techniques, challenges. Optimal design
7	Transform technique: Fourier transform, its properties, inverse Fourier transform, discrete Fourier transform, properties of DFT, circular convolution, computations for	8	Computational aspects of DFT, fast realization (FFT), phase shift

	evaluating the DFT, decimation in time and decimation in frequency FFT algorithms, discrete Hilbert transform		analysis and design of filters
8	Finite word length effect	2	Precision and word length on filter coefficient
9	Introduction to adaptive filter-LMS FIR, Wiener filter theory; multi-rate filter design-Up-sampling, down-sampling, sub-band decomposition, Applications	6	Adaptive filter design, multi-rate filter design- role and uses in applications
	TOTAL	42	

Course Outcome	Students who successfully complete the course will be able to: 1. Determine the frequency response and the z-transform of discrete-time systems. 2. Determine the discrete Fourier transform of discrete-time signals 3. Design Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters 4. Design adaptive filters pertaining to diverse applications
-----------------------	---

Learning Resources	1. Digital Signal Processing, Principles, Algorithms and Applications- Proakis, Manolakis and Sharma, Pearson Education. 2. Digital Signal Processing by Alan V. Oppenheim and Ronald W. Schaffer, Prentice-Hall. 3. Digital Signal Processing Fundamentals and Applications- Li Tan and Jean Jiang. 4. Digital Signal Processing- A computer-Based Approach- Sanjit K. Maitra, TMH. 5. Digital Signal Processing with Field Programmable Gate Arrays, U.Meyer-Baese, Springer Third Edition.
---------------------------	---

Course Code	IT 5265N	Course Name	Cyber-Physical Systems and Security	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Basic knowledge of embedded systems, operating systems, networks, and control systems.	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology	Data Book / Codes/Standards		None

Course Objectives	– To introduce fundamental concepts of cyber-physical systems (CPS) and their applications. – To understand the design, modeling, and analysis of CPS. – To analyze security challenges in CPS across different domains. – To explore emerging trends and solutions in CPS security, including real-time systems and networked embedded systems.
--------------------------	---

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Module 1: Introduction to Cyber-Physical Systems (Week 1–2) – Definition, characteristics, and components of CPS – CPS vs. Embedded Systems vs. IoT – Applications: Healthcare, Smart Grid, Industrial Control, Automotive, Robotics – CPS architecture: sensing, computation, actuation, and feedback loop	6	– Understand and articulate the definition, motivation, and characteristics of CPS. – Differentiate CPS from IoT and embedded systems. – Identify application domains of CPS.
2	Module 2: Modeling and Design of CPS (Week 3–4) – Continuous and discrete systems – Hybrid systems and state-space modeling – Tools: Simulink, Modelica, Ptolemy II – System specification: FSMs, Petri nets, hybrid automata – Control-theoretic foundations	6	– Learn system-level modeling of CPS using hybrid automata and FSM. – Understand and apply model-based design paradigms. – Use modeling tools like Simulink or Modelica in basic examples.
3	Module 3: Communication and Networking in CPS (Week 5–6) – Real-time communication constraints – Time-Triggered vs. Event-Triggered Architectures – Network topologies: ZigBee, CAN, WirelessHART, TSN – Middleware for CPS (DDS, MQTT, ROS)	6	– Understand timing constraints in CPS. – Explore scheduling algorithms and resource management in real-time systems. – Recognize hardware-software co-design principles.
4	Module 4: Security Challenges in CPS (Week 7–8) – CPS threat landscape: physical, cyber, and cyber-physical attacks – Attack surfaces: sensors, actuators, network, control – Case studies: Stuxnet, BlackEnergy, Maroochy incident – Adversary models in CPS	6	– Explain the role of communication protocols in CPS. – Analyze latency, jitter, and reliability issues. – Evaluate wired and wireless protocols (e.g., TSN, Zigbee, 6LoWPAN).
5	Module 5: CPS Security Techniques (Week 9–10) – Security principles: confidentiality, integrity, availability, resilience – Secure control design and anomaly detection – Cryptographic protocols for constrained devices – Intrusion detection and prevention systems (IDS/IPS) in CPS – Formal verification and runtime monitoring	6	– Understand threat models specific to CPS. – Explore authentication, encryption, and intrusion detection mechanisms. – Analyze case studies of real-world CPS attacks.

6	Module 6: Real-Time Operating Systems and Scheduling (Week 11) – RTOS basics and CPS-specific RTOS (FreeRTOS, VxWorks, RTEMS) – Task scheduling and resource management in CPS – Determinism and temporal correctness	3	– Learn RTOS principles, task scheduling, and resource management principles.
7	Module 7: Privacy, Ethics, and Legal Aspects (Week 12) – Privacy preservation in CPS data streams – GDPR, HIPAA, and regulatory frameworks – Ethical implications of autonomous CPS (e.g., autonomous vehicles, drones) – Risk assessment and mitigation frameworks	3	– Apply privacy-preserving techniques to CPS data handling. – Interpret relevant legal frameworks (GDPR, HIPAA) in the context of CPS. – Assess ethical dilemmas arising from autonomous CPS operation. – Conduct risk assessment and propose mitigation strategies for CPS projects.
8	Module 8: Research and Emerging Topics (Week 13–14) – AI/ML for CPS security – Digital twins and CPS simulation – Blockchain for decentralized CPS security – Quantum threats and CPS resilience – Industry 4.0, Smart Cities, and future CPS trends	6	– Evaluate AI/ML approaches for CPS security enhancement. – Describe the role of digital twins in CPS simulation and optimization. – Assess blockchain applications for decentralized CPS trust management. – Discuss the implications of quantum computing threats for CPS resilience. – Analyze Industry 4.0, smart city initiatives, and future CPS trends.
	TOTAL	42	

Course Outcomes	
------------------------	--

Learning Resources	Core Textbooks: – Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press, 2015 – Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd Edition, 2017 (Open Access) – Al-Sakib Khan Pathan, Security of Self-Organizing Networks: MANET, WSN, WMN, VANET, CRC Press Supplementary References: – CPS Public Working Group: NIST SP 1500-201 – MIT OpenCourseWare on CPS – Selected papers from IEEE Transactions on CPS, ACM TloT, and ACM CCS
---------------------------	---

Course Code	IT 5266N	Course Name	Cognitive Radio and Networks	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	The students are expected to have basic knowledge on Wireless Communications/ Wireless Networks, Communication Engineering	Co-requisite Courses	Digital Signal Processing Soft computing	Progressive Courses	5G Beyond, IoT Communications. Cognitive Radio Access Networks (CRANs)
Course Offering Department		Information Technology		Data Book / Codes/Standards	None

Course Objectives	Spectrum and energy efficient 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 potentials
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: Energy Efficient CSS, security threats in CSS, PUEA and SSDF attacks	8	Spectrum information, assessment through sensing & prediction
3	Joint SS and Data Transmission: Link layer design and common control channel, resource allocation-power allocation and channel assignment, optimized system design	8	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	6	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 in CRN data transmission: Eavesdropping and secrecy outage in CRN, Jamming for eavesdropping protection, jammer selection, ergodic capacity analysis	6	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	2	Sustainable CRN, TS, PS, SWIPT mode impact of EH
8	Application Specific System Design in CRN- Wireless Medical Telemetry Services (WMTS), cognitive radio vehicular networks (CR-VANET), CR for emergency communication, CR-IoT	6	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. 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.
-----------------------	---

Learning Resources	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	IT 5267N	Course Name	Quantum and Neuromorphic Computing	Course Category	OE	L	T	P
						3	0	0

Pre-requisite Courses	Linear algebra concepts. Probability & basic information theory. Basic digital design / computer architecture and familiarity with neural networks (ANNs). Programming in Python (recommended libraries: NumPy, PyTorch/Equivalent).	Co-requisite Courses		Progressive Courses	
Course Offering Department		Information Technology		Data Book / Codes/Standards	NA

Course Objectives	Equip students with a rigorous introductory understanding of (1) quantum computation principles and algorithms, and (2) neuromorphic computing principles, models, and hardware — enabling them to analyse, program (or simulate), and compare emerging quantum and brain-inspired computing paradigms and identify appropriate application domains.
--------------------------	--

Module	Syllabus	Duration (class-hour)	Module Outcome
1	Module 0: Foundations & review (linear algebra, probability, digital basics)	3	Student refreshes linear algebra and probability essentials needed for quantum and SNN math; gains familiarity with software toolchain.
2	Module 1: Quantum mechanics for computing & quantum information (4 weeks) Topics: Qubit formalism; Bloch sphere; single- and two-qubit gates; density matrices; measurement; tensor products; entanglement; basic quantum information measures (von Neumann entropy).	12	Represent quantum states and gates using linear algebra. Compute measurement probabilities and simple entanglement metrics.
3	Module 2: Quantum circuits & algorithms (2 weeks) Topics: Quantum circuit model; circuit identities; Deutsch-Jozsa; Bernstein-Vazirani; Grover search; (introduction to) Shor's factoring idea (conceptual). Circuit complexity basics.	6	Design and analyze small quantum circuits; understand algorithmic speedups and limits.
4	Module 3: Quantum error, noise, and NISQ era (1 week) Topics: Sources of noise; decoherence; quantum error correction basics (bit-flip, phase-flip, stabilizer idea); NISQ algorithms (VQE, QAOA concept).	3	Explain decoherence and the motivation for error correction and NISQ-era hybrid algorithms.
5	Module 4: Quantum programming & hands-on (1 week)	3	Implement, simulate and analyze small quantum algorithms using an available simulator.

	Topics: Intro to Qiskit / Cirq (or vendor SDK). Simulating circuits, running on cloud backends (concepts), writing small experiments.		
6	Module 5: Introduction to neuromorphic computing (1 week) Topics: History & motivation; brain inspiration; energy efficiency and event-driven computation; comparison to von-Neumann and deep learning accelerators.	3	Articulate why neuromorphic architectures exist and their potential advantages/limitations.
7	Module 6: Spiking neuron models & SNNs (1 week) Topics: LIF (leaky integrate-and-fire), Hodgkin–Huxley (overview), spike coding schemes, temporal coding vs rate coding, synapses.	3	Model and simulate single neurons and small networks; explain spike encoding principles.
8	Module 7: Learning in SNNs & plasticity (1 week) Topics: STDP, Hebbian rules, surrogate gradients, conversion vs native training for SNNs, on-chip learning concepts.	3	Compare learning rules and implement a basic STDP or surrogate-gradient training experiment.
9	Module 8: Neuromorphic hardware & platforms (1 week) Topics: Survey of hardware: Intel Loihi, IBM TrueNorth, SpiNNaker, memristive approaches, digital neurosynaptic chips and event-based sensors (DVS). Energy-efficiency case studies.	3	Identify characteristics of leading neuromorphic platforms and map workloads where they excel.
10	Module 9: Comparative applications, future directions & student presentations (1 week) Topics: When to use quantum vs neuromorphic approaches (optimization, sensing, cryptography, low-power inference); ethics, sustainability, research frontiers. Student project demos & course recap.	3	Critically evaluate and present a comparative analysis or prototype project.
	TOTAL	42	

Course Outcomes	After completing the course, students will be able to: Explain core quantum computation concepts (qubits, gates, circuits, entanglement) and basic quantum algorithms. Formulate and analyse simple quantum circuits and reason about resource counts (qubits, depth, gate counts). Describe neuromorphic principles (spiking neuron models, plasticity rules) and compare SNNs vs ANN implementations. Use simulation tools to prototype simple quantum circuits (e.g., Qiskit / Cirq) and spiking neural networks (e.g., Brian2, Nengo, Loihi SDK). Critically evaluate strengths, limitations, and application fit of quantum vs neuromorphic approaches for candidate problems (optimization, sensing, low-power edge AI).
------------------------	--

Learning Resources	Primary (core) – M. A. Nielsen & I. L. Chuang, Quantum Computation and Quantum Information — canonical textbook for quantum computation theory. – Course notes / modern course syllabi (examples from Purdue and others) for quantum course structure and pedagogy. Neuromorphic & SNNs – Course syllabi and teaching material from neuromorphic courses (e.g., SDSU, CMU, other universities). Useful for lab structure and hardware survey. – Intel Loihi, SpiNNaker and other platform documentation / vendor SDKs (use for hands-on labs where accessible). Supplementary / approachable reads – S. Aaronson, Quantum Computing Since Democritus (intuitive supplement). – Recent review papers and arXiv course-notes about teaching quantum computing and neuromorphic tech (great for updated labs & assignments)
---------------------------	---