Indian Institute of Technology Jodhpur

2nd year B.Tech Computer Science & Engineering
Curriculum Structure
August 2020

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

2^{nd} year B.Tech Computer Science & Engineering
Curriculum Structure

1. Introduction

Traditional curriculum in Computer Science and Engineering (CSE) has focused on two major areas, i.e., theory and systems (database, computer hardware, and operating systems). With the advent of the era of Edge Computing, Cloud Computing, Security, AI and Big Data, the discipline is being transformed by the incorporation of new emerging technologies. The discipline is becoming instrumental in solving major real-world problems faced by modern society such as healthcare, security, sustainability, and socio-economic challenges. There is a need to revamp the B.Tech program in CSE so as to prepare students to tackle problems which are fundamentally multidisciplinary in nature. This would improve IIT Jodhpur's participation and contribution towards providing solutions for global challenges. IIT Jodhpur can become a leading source of technology solutions for self-reliant India.

2. Objectives of the Program

The objective of the B.Tech. program in Computer Science and Engineering (CSE) at IIT Jodhpur is to equip students to undertake careers involving challenges of working on real-world problems, while innovating with their core competency in computer science. The curriculum gives due importance to the foundational aspects of computer science, as well as develops in students the necessary engineering skills for addressing emerging technological challenges. The program offers flexibility in choosing elective courses for widening the understanding of emerging concepts and processes in different domains. Students can opt for specializations which provide a coherent and increasingly sophisticated understanding so as to enable them to pursue preferred career opportunities. The program will produce a well prepared and well-motivated workforce to undertake careers in research and industry involving innovation, knowledge creation, engineering, and entrepreneurship.

3. Expected Graduate Attributes

After completing this program, a student will be able to develop an ability to:

1. Apply appropriate knowledge in Computer Science and Engineering to identify, formulate, analyze, and solve complex engineering problems in order to reach substantive conclusions.
2. Self-learn and engage in use of advanced computing tools.
3. Develop sustainable computing solutions in broader economic, societal and environmental contexts.
4. Think critically, creatively and analytically as a computer scientist, whilst being able to work effectively, independently and collaboratively as part of a team in research, technology development and entrepreneurial ventures.
5. Apply evolving ethics and privacy laws across various domains and territories.
6. Effectively communicate engineering concepts and ideas to peers in written or oral forms.
7. Be motivated to engage in independent and life-long learning in the broadest context of evolving technological challenges.

4. Learning Outcomes

The student will have the ability to:

1. Abstract out computational models based on mathematical rigor.
2. Critically analyse and appreciate emerging and applied fields of computer science owing to deeper understanding and strong foundations in theoretical computer science.
3. Develop robust, secure, and efficient software systems that would involve software engineering, data structures, and algorithm design paradigms.
4. Apply different concepts including digital design, computer architecture, operating systems, databases, networking, and applications for building solutions for real-world problems.
5. Design and implement smart, intelligent, and user friendly interfaces for computer applications.
6. Analyse, design, develop and deploy computer based systems in collaboration with domain experts for meeting societal needs in areas such as healthcare, safety, environment, law enforcement, transportation, etc., by exploiting tools and techniques of Big data, AI and project management.
7. Become proficient in some of the specialized areas such as Computer Vision, Augmented Reality, Internet of Things, Natural Language Understanding, and Intelligent Communication.

8. Communicate effectively by comprehending, documenting, making effective presentations and exchanging clear instructions.

5. New Skill Sets Targeted

1. Designing user-centric human-machine interfaces
2. Designing end-to-end socio-digital reality systems
3. Developing sustainable computing solutions
4. Building secure, intelligent, and trustable systems
5. Designing and synthesizing digital systems for solving industrial problems

6. Topic Clouds and Mapping of Topic Clouds with Proposed Courses

Table 1. Topics and Mapping of Topic with Courses

Area	**Topics**	**Category (Core/ Techniques Technology/ Systems)**	**Course (IE/IS/ PC/PE)**
Theory and Algorithms	Computing Paradigms, Syntax, Semantics Specification, Attribute Grammar, Imperative Programming, Procedures, Modules, Object-oriented Programming, Abstraction, Encapsulation, Reuse, Functional Programming, Polymorphic Data Types, Recursion, Lambda Calculus, Logic Programming, Horn Clauses, Backtracking, Concurrent Programming, Synchronization, Multithreads	Core	Principles of Programming Languages (PC)
	Abstract Data Types, Linear Data Structures, Non Linear Data Structures, Stack, Queue, Link List, Heap, Sorting, Hashing, Algorithm Analysis, Graph, Tree	Core	Data Structures and Algorithms (PC)
	Complexity Analysis, Divide-and-conquer, Greedy Algorithms, Dynamic Programming, Linear Programming, Universal Hashing, Approximation Algorithms, Randomized Algorithms	Core	Design and Analysis of Algorithms (PC)
	Discrete Structures, Logics, Set, Graph and Trees, Deterministic Finite Automata, Non-deterministic Finite Automata, PushDown Automata, Context Free Grammar, Turing Machine, Lexical and Syntax Analysis, Parsing	Core	Maths for Computing (PC)
	Computability, Decision Trees and Communication Complexity, Time Complexity, Space Complexity, Complexity of Counting and Randomization	Core	Theory of Computation (PE)
	Graph algorithms, Directed and Undirected graph, Planar graph, Graph coloring, Hamiltonion and Eulorian graph, Bipartite graphs, Trees.	Core	Graph Theoretic Algorithms (PC Bouquet)
	NP-Hard problems, Bounded Search Tree, Iterative Compression, Randomized Methods, Kernelization, Sunflower Lemma, Crown Decomposition, Expansion Lemma, Intractability	Core	Parameterized Complexity (PC Bouquet)

	Persistence DS, Retroactive DS, Geometric DS, DS for Moving Data, Dynamic Optimality, Cache-oblivious DS, van Emde Boas, Sketching, String DS, Low Memory DS, Dynamic Graphs	Core and Techniques	Advanced Data Structures (PE)
	Euclidean TSP, Metric TSP, FPTAS, PTAS, Primal-Dual Schema, Dual Fitting, Embeddings, Sparsest Cut, Random Hyperplanes, Hashing, Makespan	Techniques	Approximatio n Algorithms (PE)
	Randomized Complexity Classes, Probabilistic Analysis, Markov Chain, Random Walk, Randomized Graph Algorithms, Randomized Sorting and Searching, Markov, Chebyshev, Chernoff	Techniques	Randomized Algorithms (PE)
	Computational Model and Complexity Classes, NP, NP Completeness, Complexity Issues in Cryptography, Lower Bounds, Algebraic Computational Model, Levin's Theory, Hardness Amplification, Logic in complexity theory, Circuit Lower Bounds	Core	Complexity Theory (PE)
	Complex Numbers, Complex Vector Spaces, Quantum Algorithms, Quantum Programming Languages, Quantum Cryptography, Quantum Information Theory, Deutsch's Algorithm, Simon's Periodicity Algorithm, Grover's Search Algorithm, The Deutsch–Jozsa Algorithm, Shor's Factoring Algorithm, Quantum Assembly Programming, Quantum Key Exchange, Quantum Teleportation, von Neumann Entropy	Core	Quantum Computing (PE)
	Streaming Algorithms, Stream mining using Clustering, Massive Data Clustering, Data Stream Classification, Distributed Mining of Streaming Data, Change Diagnosis, Forecasting on Stream, Dimensionality Reduction for Streaming data.	Core and Technique	Stream Analytics (PE)
Hardware	Introduction to HDL, Design flow, Number representation, Arithmetic circuits, Design and Analysis of Synchronous circuits, Design of Finite State Machines, CMOS Technology, Programmable Logic Devices, CAD	Core	Digital Design (PC)
	CPU, Memory, Processor, Instruction set Architecture, I/O, ALU, Bus Architecture, Cache management, Performance	Core	Computer Architecture (PC)
	Embedded system design, Architectures of embedded processors, Hardware-Software Co-design, ARM Cortex M3, Programming for Embedded Systems, Real time OS, Distributed Embedded Architectures, Real time scheduling algorithms, Multi-rate systems	Systems	Embedded Systems (PC Bouquet)
	VLSI design flow, hardware modelling principles, hardware description, Behavioural synthesis, logic optimization and synthesis, Chip architectures, standard cells and FPGA, Synthesis tools, layout synthesis	Technique	High-level Synthesis (PE)

Systems	Processes Management, Process Coordination, Deadlock, Memory Management, Storage Management, Protection and Security	Core	Operating Systems (PC)
	Database Architectures, Data Granularity, ER Model, EER Model, Relational Model, Relational Algebra, Relational Calculus, SQL, Schema, Query, Data Definition, Data Manipulation, Data Retrieval, Views, Normalization, Functional Dependency, Transactions, Schedules, ACID Properties, Concurrency Control, Recoverability, Serializability, Deadlock Prevention, Locking Protocols, Timestamp Protocols, Deadlock Detection, Recovery Protocols, Query Optimization, Hashing, Indexing	Core	Database Systems (PC)
	Software Development Life Cycle, Software Engineering Process Model, Software Design, UML, Design Pattern, Project Management, Cost Analysis, UI Design, Service Oriented Architecture, Software Quality, Risk Analysis, Agile Methodologies, Software Architecture	Core	Software Engineering (PC)
	Software Testing, Quality Engineering, Testing Process, Automatic Testing, Unit Testing, Defect Prevention, Process Improvement, Quantifiable Quality Improvement, Quality Model and Measurements, Defect Classification, Reliability Engineering	Techniques	Software Testing and Quality Assurance (PE)
	Distributed Systems, Centralized Systems, Data Transparency, Data Fragmentation, Data Allocation, Security, Semantic Integrity, Query Decomposition, Query Localization, Query Ordering, Query Optimization, Transaction Management, Concurrency Control, Deadlock Management, Reliability, Recovery Protocols	Systems	Distributed Database Systems (PE)
	Fog Computing, Internet of Things and Analytics, Edge Computing, Distributed computing and system design, architecture, contemporary cloud framework, distributed data & processing management, middleware design issues and challenges, Edge & Fog for AI-driven applications	Systems	Edge and Fog Computing (PE)
	Concurrency, events & ordering, distributed file systems, distributed disk management, fault tolerance, microkernel, operating system kernel	Systems	Distributed Systems (PE)
	Understanding Software & Evolution, Software Maintenance, Software Reuse, Reverse engineering levels & techniques	Systems	Software Maintenance (PE)
Network and Security	Cyber Security Vulnerabilities and Cyber Security Safeguards, Securing Web Application, Services and Servers, Intrusion Detection and Prevention, Firewalls, Cyberspace and the Law, Cyber Forensics	Core	Cyber Security (PC)
	OSI reference model, Packet switching techniques, Performance metrics, Socket programming, Email, FTP, Telnet, SSH, DNS, TCP, UDP, Transmission, Flow and Congestion Control, Tunneling, Internet Protocol and its operation, Routing algorithms, ICMP, ARP, RARP, DHCP, IPv6, RIP,	Core	Computer Networks (PC)

	OSPF, Advanced Internetworking, Multicast routing, framing, medium access mechanism, Public key and private key cryptography, Firewalls, SDN Overview.		
	Overview about MANETs, VANETs, WSNs and WMNs, Modulation techniques at PHY layer, Fading mitigation techniques, MAC protocols for MANETs, Error correcting codes, distributed wireless routing algorithms, transport level mobility management, 5G overview, LPWAN, Bluetooth	Systems	Introduction to Wireless Ad hoc Network (PE)
	Digital Trust, Digital Asset, Digital Transactions, Distributed Ledger Technology, Digital Signature, Cryptocurrency, Bitcoin, Hyperledger, Ethereum Interaction, Blockchain-as-a-service, Blockchain Use Cases, Blockchain Security	Technology	Introduction to Blockchain (PE)
	Cooperative Vehicular Safety Applications, Vehicular Mobility Modeling, Physical Layer Considerations for Vehicular Communications, MAC Layer of Vehicular Communication Networks, VANET Routing protocols, Emerging VANET Applications, Standards and Regulations	Systems	Vehicular Ad-Hoc Networks (PE)
	Overview of wireless systems, Multiplexing, Link adaptation, Multihop routing protocols, Mobility and Handoff management, Cellular Technologies, Overview of LTE and 5G Cellular Networks, 5G Architecture, RAN and dynamic CRAN, Mobility management and Network slicing in 5G, Pervasive computing, Context aware sensor networks, Localization, Android architecture and Overview, Body area networks	Technology	Mobile and Pervasive Computing (PE)
	4G LTE networks, From 4G to 5G, 5G overview, core, signalling, 5G mobile edge and fog computing, application, 5G vs mm-wave WiFi, India's 5G policy and vision, 6G vision, paper discussions	Technology	5G Mobile Networks (PE)
	Networking basics, switching architecture, SDN architectures, OpenFlow, Network function virtualization, Emerging SDN models, Data center networking	Core	Software Defined Networks (PE)
	Basic symmetric-key encryption, Message integrity, Public key cryptography, Digital signatures ,Protocols	Core	Cryptography and Network security (PE)
Applications	Bayes Decision Theory, Regression, Bias variance, Maximum Likelihood Estimation, Bayesian Parameter Estimation, Decision Tree, Random Forest, Artificial Neural Network, Clustering, k-means, SVM, Feature Selection, Dimensionality Reduction	Core and Technique	Pattern Recognition and Machine Learning (PC)
	Uninformed Search Strategies, Informed Search Strategies, Local Search Algorithms, Hill Climbing, Constraint Satisfaction Problems, Backtracking, Adversarial Search, Min-Max algorithms, Propositional Logic, Reasoning Patterns, First-order logic, Syntax, Semantics, Q-value, Policy,	Technique	Artificial Intelligence (PE)

	Search Engine Architecture, Retrieval Models, Performance Evaluation, Text Categorization, Text Clustering, Web Information Retrieval, Structured Document Retrieval	Techniques	Information Retrieval (PE)
	Visual World, Geometry, Lights and Optics, Tracking, Motion, Depth, Devices and tools	Technique	Introduction to AR and VR (PE)
	Geometric primitives, Clipping, Viewing, Rendering, Animation, Shading, Coloring, OpenGL	Technique	Computer Graphics (PE)
	Eye and early vision, Reasoning systems, Late vision, visual attention, Cognitive architectures, Knowledge representation and learning, Neural networks for vision, Applications in vision tasks	Technique	Principles of Biological Vision and Applications (PE)
	Image formation and transformations, Camera calibration, Image restoration, Spatial and Wavelet-based processing, Epipolar Geometry, SfM, Optical flow, Key-point detection, Feature description and matching, Deep learning for vision, Applications	Technique	Computer Vision (PE)
	Data storage methods, data access methods, analysis, representation, visualization, visual storytelling, Dashboard, Infographics	Technique	Data Visualization (PE)
	Neural Networks, Gradient Descent, Optimization, Regularization, Autoencoder, Convolutional Neural Network, Recurrent Neural Network, LSTM, Deep Generative Models, Generative Adversarial Network (GAN), Deep Belief Network, Deep Convolutional GAN, Variational Autoencoder, Representation Learning, Unsupervised Pre-training, Transfer Learning, Distributed Representation, Domain Adaptation, Neural Language Model, Adversarial Learning	Technique	Deep Learning (PE)
	Kernel Machines, Variants of Support Vector Machines, PAC Theory, Boosting, Graphical Models, Structural Predictions, Deep Reinforcement Learning, Space Coding	Core and Technique	Advanced Machine Learning (PE)
	Decision-making, Utility Theory, Utility Functions, Decision Networks, Sequential Decision Problems, Partially Observable MDP, Game Theory, Reinforcement Learning, Generalization, Policy Search, Hidden Markov Model, Kalman Filter, Knowledge Representation, Ontological Engineering, Situation Calculus, Semantic Networks, Description Logic, Planning graphs, Partial-order Planning, Conditional Planning, Continuous Planning, Multi-agent Planning, Hierarchical Task Network Planning, Non-deterministic Domains	Core	Advanced Artificial Intelligence (PE)
	Neuromorphic Engineering, Neuroanatomy of human brain, Neuron models, Plasticity rules, Learning rules, Spiking Neural Networks (SNN), Nanodevices for Neuron Implementation, Synaptic emulation, Electronic synapses, Digital/Analog neuromorphic VLSI, Hardware implementation of synaptic circuits, Synaptic programming	Technique	Neuromorphic Design and Computing (PE)

	Accuracy-explainability tradeoff, Interpretability problem, Predictability, Transparency, Traceability, Causality, Reasoning, Attention and Saliency, Interpretable AI, Prediction Consistency, Adversarial Robustness, Trustworthy AI, Integrity, Reproducibility, Accountability, Bias-free AI, Ethics in AI	Technique	Dependable Artificial Intelligence (PE)
	Fuzzy Computing, Fuzzy Systems, Neural Computing, Genetic Algorithms, Rough Sets, Knowledge Representations, Fuzzification, Defuzzification, Swarm Intelligence, Hybrid Models, Neuro-fuzzy systems, Rough-neural computing	Technique	Soft Computing Techniques (PE)
	Biological Signals, Biomedical Imaging Modalities, ECG, NMR spectroscopy, electron microscopy, MRI, and X-Ray Images, Visualization, Reconstruction, Restoration, Clustering, Graph Partitioning, Deep Learning for Bio-image	Technique	Bio-image Computing (PE)
	Biometric Devices, Biometric System Design, Biometric Data Analysis, Face Recognition, Fingerprint Matching, Signature Authentication, Biometric	Technique	Advanced Biometrics (PE)
	Computational complexity of AI models, Prediction accuracy, Numeric accuracy, Precision, Memory footprints, Edge AI, Memory Optimization of Models, Hardware accelerators for Edge AI, Vision Processing Unit, Streaming Hybrid Architecture Vector Engine, Open Neural Network Exchange	Systems and Technique	Resource-Constrained Artificial Intelligence (PE)
	Word representation, NLP tasks, Seq2Seq model, Question Answering, Sentiment Analysis, Dialogue system, Machine Translation, natural language generation Interpretability, Knowledge Graphs.	Technique	Natural Language Understanding (PE)
	Spoken language technology, dialog and conversational systems, automatic speech recognition, speech synthesis, affect detection, dialogue management.	Techniques	Speech Understanding (PE)
	Graphs, Network Models, Network Data Generation, Structural Properties, Link Prediction, Community Detection, Information Cascade, Small World Phenomenon, Homophily, Structural Balance, Components, Network Evolution, Multi-layer network	Technique	Social Networks (PE)

7. Course Categories, Credit Distribution and Credit Structure of B.Tech. Programmes

Table 2. Proposed Course Categories and credit distribution in the proposed B.Tech. Programmes

S.N.	Course Type	Course Category	Regular B.Tech.		Double B.Tech.	
			Credit	Total	Credit	Total
1	Institute Core (I)	Engineering (IE)	34	69	34	59
		Science (IS)	16		16	
		Humanities (IH)	12		9	
2	Programme Linked (L)	Science (LS)	7		0	
3	Programme Core (P)	Programme Compulsory (PC)	51	71	51	71
		Programme Electives (PE)	17		17	
		B.Tech. Project (PP)	3		3	
4	Open (O)	Open Electives (OE)	10	10	0	0
5	Engineering Science (E)	Engineering Science Core (EC)	0	0	22	22
		Engineering Science Elective (EE)	0	0	8	8
Total Graded				**150**		**160**
6	Non-Graded (N)	Humanities (NH)	6	15	6	15
		Engineering (NE)	3		3	
		Design/Practical Experience (ND)	6		6	
Total Graded + Non-Graded				**165**		**175**

8. Credit Structure of B.Tech. Programmes

Table 3. Credit Structure for B.Tech. Programmes (Up 6000 Level)

Type	L-T-P	Distribution of contact and beyond contact hours			Total Credits (TC=TH/3)
		Contact Hours (CH)	Beyond Contact Hours (BCH)	Total Hours (TH)	
1 hour of Lecture	1-0-0	1 hr	2 hr	3 hr	1
1 hour of Tutorial	0-1-0	1 hr	2 hr	3hr	1
1 hour of Lab/Project	0-0-1	1 hr	0.5 hr	1.5 hr	0.5

#Contact hour for a project refers to the involvement of students in the laboratory, discussion, etc.

9. List of Programme Compulsory Courses

Table 4. Programme Compulsory Courses

Sr. No	Course Name	LTP	Contact Hours	Credit
1	Data Structure and Algorithms	3-0-2	5	4
2	Maths for Computing	3-1-0	4	4
3	Digital Design	2-0-2	4	3
4	Human-Machine Interaction	0-0-4	4	2
5	Software Engineering	3-0-2	5	4
6	Operating Systems	3-0-2	5	4
7	Design and Analysis of Algorithms	3-1-0	4	4
8	Computer Architecture	3-0-0	3	3
9	Database Systems	3-0-2	5	4
10	Cyber Security	3-0-2	5	4
11	Computer Networks	3-0-3	5	4.5
12	Digital Systems LAB	0-0-3	3	1.5
13	Principles of Programming Languages	3-0-0	3	3
14a	Randomized Algorithms (B1)	3-0-0	3	3
14b	Approximation Algorithms (B1)	3-0-0	3	3
14c	Distributed Algorithms (B1)	3-0-0	3	3
14d	Optimization (B1)	3-0-2	5	4
15a	VLSI Design (B2)	3-0-0	3	3
15b	High-Level Synthesis (B2)	3-0-0	3	3
15c	Embedded Systems (B2)	3-0-0	3	3
			Total	**51/52**

10. Area-wise Programme Elective Courses

Table 5. Stream-wise Programme Electives Courses

S. No.	Stream	Courses	L-T-P
1	Theory and Algorithms	1. Randomized Algorithms (700) 2. Approximation Algorithms (700) 3. Complexity Theory (400) 4. Advanced Data Structures (400) 5. Quantum Computing (400) 6. Graph-theoretic Algorithms (700) 7. Parameterized Complexity (700) 8. Optimization (Math) (400)	3-0-0
2	Hardware	1. High-level Synthesis (EE/CS)(700) 2. Embedded Systems (EE) 3. VLSI Design (EE) 4. Hardware Software Co-Design (CS) (700)	3-0-0
3	Systems	1. Software Testing and Quality Assurance (700) 2. Distributed Database Systems (700) 3. Distributed Systems (2-0-0) (400) 4. Mobile and Pervasive Computing (700) 5. Virtualization and Cloud Computing (700) 6. Compiler Design (700) 7. Software Maintenance (400)	3-0-0 2-0-2
4	Network and Security	1. Introduction to Wireless Ad hoc Network (700) 2. 5G Mobile Networks(CS)(700) 3. Software Defined Networks (CS)(700) 4. Intelligent Radio Networks (EE) (700) 5. Data Communication Networks (EE) (300) 6. Cryptography and Network Security (CS/EE) (400) 7. Introduction to Blockchain (700)	3-0-0 2-0-0 3-0-2
5	AI-ML	1. Deep Learning (400) 2. Artificial Intelligence (400) 3. Advanced Machine Learning 4. Dependable Artificial Intelligence (400) 5. Information Retrieval (700) 6. Autonomous Systems (IDRP) (400) 7. Ethics, Policy, Law and Regulation in AI (700) 8. Planning and Decision making for Robots (IDRP)(700)	3-0-0
6	Speech,Text, and Vision	1. Natural Language Understanding (700) 2. Digital Image Processing (400) 3. Computer Vision (400) 4. Computer Graphics (400) 5. Speech Understanding (700)	3-0-0

7	Social Computing	1. Social Computing 2. Sustainable Computing 3. ICT for Development 4. Machine Learning for Epidemiology 5. Health Informatics 6. Computational Social Choice 7. Computational Cognition & Behavior Modelling 8. Social Networks 9. Ethics, policy, law and regulations in AI 10. Crowd-sourcing and human computing 11. Environmental Informatics 12. Computational Microeconomics	3-0-0 3-0-0 2-0-0 3-0-0 3-0-0 3-0-0 3-0-0 3-0-0 0-0-2 3-0-0 3-0-0 3-0-0

11. Specializations Offered by the Department

Table 6. Specializations and courses

S. No.	Name of Specialization	Specialization Core (8 Credits)	Specialization Elective (12 Credits)
1.	Visual Computing (CSE, AIDE, EE)	Jointly with EE and AIDE	Jointly with EE and AIDE
2.	Socio-Digital Reality (CS, AIDE, EE)	Jointly with EE and AIDE	Jointly with EE and AIDE
3.	Advanced Algorithms	Advanced Data Structures (3-0-0), Complexity Theory (3-0-0), Algorithms for Big Data (from M.Tech, 2-0-0)	Randomized Algorithms, Approximation Algorithms, Theory of Computation, Distributed Algorithms, Quantum Computing (IDRP), Optimization, Graph Theory and Applications, Parameterized Complexity
4.	Social and Sustainable Computing	Social Computing (3-0-0), Sustainable Computing (3-0-0), ICT for Development (2-0-0)	Machine Learning for Epidemiology (3-0-0), Health Informatics (3-0-0), Computational Social Choice (3-0-0), Computational Cognition & Behavior Modelling (3-0-0), Social Networks (3-0-0), Ethics, Policy, Law and regulations in AI(0-0-2), Crowd-sourcing and human computing (3-0-0), Environmental Informatics (3-0-0), Computational Microeconomics (3-0-0), Project (0-0-6)
5.	Intelligent Communication and Networking (CS, AIDE, EE)	Jointly with EE	Jointly with EE
6.	VLSI Systems (EE, CS)	Jointly with EE	Jointly with EE

12. Curriculum of B.Tech. in Computer Science and Engineering (Direct Entry)

Table 7. Curriculum of B.Tech. (Computer Science and Engineering)

Cat	*Course*	*LTP*	*CH*	*NC*	*GC*	*Cat*	*Course*	*LTP*	*CH*	*NC*	*GC*
	I Semester						**II Semester**				
IE	Introduction to Electrical Engineering	3-0-2	5	-	4	IE	Engineering Mechanics	2-1-0	3	-	3
IE	Introduction to Computer Science	3-0-2	5	-	4	IS	Chemistry	3-0-0	3	-	3
IE	Introduction to Bioengineering	3-0-2	5	-	4	IS	Physics	3-0-0	3	-	3
						IS	Chemistry Lab	0-0-2	2	-	1
						IS	Physics Lab	0-0-2	2	-	1
IS	Mathematics I	3-1-0	4	-	4	IS	Mathematics II	3-1-0	4	-	4
IE	Engineering Visualization	0-0-2	2	-	1	IE	Engineering Realization	0-0-2	1	-	1
NE	Engineering Design I	0-0-2	2	1	-	NE	Engineering Design II	0-0-2	2	1	-
NH	Communication Skill I	0-0-2	2	1	-	NH	Communication Skill II	0-0-2	2	1	-
NH	Social Connect and responsibilities I	0-0-1	1	0.5	-	NH	Social Connect and responsibilities II	0-0-1	1	0.5	-
NH	Performing Arts I /Sports I	0-0-1	1	0.5	-	NH	Performing Arts II/Sports II	0-0-1	1	0.5	-
	Total	**12-1-14**	**27**	**3**	**17**		**Total**	**11-2-12**	**24**	**3**	**16**
	III Semester						**IV Semester**				
IS	Probability, Statistics and Stochastic Processes	3-1-0	4	-	4	IE	Materials Science & Engineering (Electronic materials)	1-0-0	1	-	1

IE	MSE - (i) Energy materials (ii) Computational Material Design	2X 1-0-0	2	-	2	IE	Thermodynamics	3-1-0	4	-	4
IE	Signals and Systems	3-1-0	4	-	4	IE	Pattern Recognition and Machine Learning	3-0-2	5	-	4
PC	Maths for Computing	3-1-0	4		4	PC	Software Engineering	3-0-2	5		4
PC	Data Structures and Algorithms	3-0-2	5	-	4	PC	Human-Machine Interaction	0-0-4	4		2
LS	Foundations of Quantum Information Processing (Program Linked Science Elective)	3-0-0	3	-	3	PC	Digital Design	2-0-2	4		3
NE	Intro. To Profession	0-0-2	2	1		IH	Humanities I	3-0-0	3		3
	Total	**17-3-4**	**24**	**1**	**21**		**Total**	**15-1-10**	**23**	**-**	**21**
	V Semester						**VI Semester**				
PC	Design and Analysis of Algorithms	3-1-0	4		4	PC	Cyber Security	3-0-2	5		4
PC	Computer Architecture	3-0-0	3		3	PC	Computer Networks	3-0-3	6		4.5
PC	Operating Systems	3-0-2	5		4	PC	Digital Systems Lab	0-0-3	3		1.5
PC	Principles of Programming Languages	3-0-0	3		3	PC	**CS Core Bouquet -1 (Algorithms)/ 2 (Hardware)**	3-0-0	3		3
PC	Database Systems	3-0-2	5		4						
IH	Humanities II	3-0-0	3	-	3	PE	Programme/Open Elective	6-0-0	6		6
NH	Professional Ethics I	0-1-0		1	-	NH	Professional Ethics II	0-0-2		1	-
	Total	**18-2-4**	**23**	**1**	**21**		**Total**	**15-1-8**	**23**	**1**	**19**
	VII Semester						**VIII Semester**				

PP	B. Tech. Project	0-0-6	6	-	3	IH	Humanities IV	3-0-0	3	-	3
PC	**CS Core Bouquet-1 (Algorithms)/2 (Hardware)**	3-0-0	3		3						
PE / OE	Programme/Open Electives	6-0-0	6	-	6	PE / OE	Programme/Open Electives	15-0-0	15	-	15
IH	Humanities III	3-0-0	3	-	3						
IS	Environmental Science	2-0-0	2	-	2						
	Total	**14-0-6**	**20**	**-**	**17**		**Total**	**18-0-0**	**18**	**-**	**18**
								Total of Graded Credits		**-**	**150**
								Total of Non-Graded Credits		**9**	**-**
								Non-Graded Design Credits		**6**	**-**
								Grand Total		**165**	

13. Curriculum of Double B.Tech.: B.Tech. in Computer Science and Engineering and B.Tech. in Engineering Science

Table 8. Programme structure of Double B.Tech.

Cat	*Course*	*LTP*	*CH*	*NC*	*GC*	*Cat*	*Course*	*LTP*	*CH*	*NC*	*GC*
	I Semester						**II Semester**				
	First two semesters same as Table 8a or 8 b 33 Graded and 6 non graded credits										
	III Semester						**IV Semester**				
ES	Probability, Statistics, and Stochastic Processes	3-1-0	4	-	4	IE	Materials Science & Engineering	3 × 1-0-0	3	-	3
ES	Modern Physics	3-0-0	3	-	3	ES	Embedded Systems and IoT	3-0-2	5	-	4
IE	Thermodynamics	3-1-0	4	-	4	IE	Pattern Recognition and Machine Learning	3-0-2	5	-	4
ES	Data Structures and Algorithms	3-0-2	5	-	4	ES	Design of Experiments	3-0-0	3	-	3
IE	Signals and Systems	3-1-0	4	-	4	ES	Control Systems	3-0-2	5	-	4
NE	Intro. To Profession	0-0-2	2	1		IH	Humanities I	3-0-0	3	-	3
	Total	**15-3-4**	**22**	**1**	**19**		**Total**	**18-0-6**	**24**	**-**	**21**
	V Semester						**VI Semester**				
PC	Design and Analysis of Algorithms	3-1-0	4		4	PC	Cyber Security	3-0-2	5		4
PC	Principles of Programming Languages	3-0-0	3		3	PC	Computer Networks	3-0-3	6		4.5
PC	Operating Systems	3-0-2	5		4	PC	Software Engineering	3-0-2	5		4
PC	Database Systems	3-0-2	5		4	PC	Human-Machine Interaction	0-0-4	4		2

PC	Maths for Computing	3-1-0	4		4	PC	Digital Design	2-0-2	4		3
						PC	**CS Core Bouquet - 1 (Algorithms)/ 2 (Hardware)**	3-0-0	3		3
IH	Humanities II	3-0-0	3	-	3						
NH	Professional Ethics I	0-1-0		1	-	NH	Professional Ethics II	0-1-0		1	-
	Total	**18-3-4**	**24**	**1**	**22**		**Total**	**15-1-14**	**29**	**1**	**20.5**
	VII Semester						**VIII Semester**				
PP	B. Tech. Project	0-0-6	6	-	3	PE / ES	Programme/Engineering Science Electives	17-0-0	17	-	17
PC	**CS Core Bouquet - 1 (Algorithms)/ 2 (Hardware)**	3-0-0	3		3	IH	Humanities III	3-0-0	3	-	3
PE / OE	Programme/ Engineering Science Electives	12-0-0	12	-	12	PC	Digital Systems Lab	0-0-3	3		1.5
PC	Computer Architecture	3-0-0	3		3						
IS	Environmental Science	2-0-0	2	-	2						
	Total	**20-0-6**	**26**	**-**	**23**		**Total**	**20-0-0**	**20**	**-**	**21.5**
									Total Graded Credits	**-**	**160**
									Total of Non-Graded Credits	**9**	**-**
									Non-Graded Design Credits	**6**	**-**
									Grand Total	**175**	

Note: ES are proposed Engineering Science compulsory courses

14. Detailed Course Content of Programme Compulsory Courses

Course Title	**Operating Systems**	Course No.	**CS XXX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-2 [4]
Offered for	B.Tech	Type	Compulsory
Prerequisite	Data Structures and Algorithms		

Objectives
To learn about design principles of operating systems
To do a case study of Operating System

Learning Outcomes
Ability to modify and compile OS
Ability to solve synchronization problems in Operating Systems

Contents
Overview of Operating Systems: Types of Operating Systems, System calls and OS structure. (4 Lectures)
Processes Management: Process, Threads, CPU scheduling. (6 Lectures)
Process Coordination: Mutual exclusion, Mutex implementation, Semaphores, Monitors and condition variables, Deadlocks. (10 Lectures)
Memory Management: Swapping, Paging, Segmentation, Virtual memory, Demand paging, Page Replacement algorithms. (8 Lectures)
Storage Management: I/O devices and drivers, Disks and file Systems, File layout and directories, File system performance, File system reliability. (8 Lectures)
Protection and Security: System protection, System security. (6 Lectures)

Laboratory
Designing a shell in Linux
Multithreaded programming using pthread
Solving the Sleeping-Barber problem
Modification of scheduling algorithm in Linux
Solving the Producer-Consumer problem over a network
Finding text, data, and stack segments of a process in Linux
Implementation of page replacement algorithms
Changing file attributes in Linux
Implementing an encrypted file system in Linux
Implementing symbolic links in Linux

Text Book
A. SILBERSCHATZ, P.B. GALVIN, G. GAGNE (2018), Operating System Concepts, John Wiley & Sons Inc., 10th Edition.

Reference Book
A.S. TANENBAUM, A.S. WOODHULL (2006), Operating Systems Design and Implementation, Pearson, 3rd Edition.
W. STALLINGS (2017), Operating Systems Internals and Design Principles, Pearson, 9th Edition.

Course Title	**Human-Machine Interaction**	Course No.	**CS2xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	0-0-4 [2]
Offered for	B.Tech (CSE, AIDE)	Type	Compulsory
Prerequisite	None		

Objectives

To provide a theoretical and practical understanding of human-machine interaction (HMI) design, including concepts of user centered and design thinking, usability, interfaces, rapid prototyping, and evaluation.

Learning Outcomes

1. The students will have:
2. A broad understanding of human-machine interaction and latest technologies.
3. Understanding of perceptual and cognitive basis of human-machine interaction.
4. Knowledge of user centered design and techniques for rapid prototyping.
5. Knowledge of assessing usefulness and usability of a design
6. Introduction to approaches for gathering and analyzing interaction data, and conveying design concepts.

Lectures (discussions in the lab and web-based flip pedagogy)

1. Theories and Research Methods in HMI
2. Understanding User Experience
3. Foundations and Concepts of Interaction
4. Prototyping Techniques
5. Prototyping Evaluation
6. Communicating User Needs and Requirements
7. Speculative Design
8. Case Studies

Laboratory and Assignments (primary approach)

1. Find a poorly designed item (anything). Submit either a picture or sketch and describe why it is poorly designed. (Week 1)
2. Heuristic Evaluation and Interview (Week 1)
3. User Scenarios, Personas and Storyboards ((Week 2)
4. User Journeys (Week 3)
5. Wireframes: Paper and Digital Prototyping (Week 4)
6. Prototype Evaluation Study Design (Week 5, 6)
7. Value Sensitive Design Evaluation (Week 7, 8)
8. Design visual Interfaces (laptop, mobile) - e.g. gesture based (Week 9, 10)
9. Design voice interfaces - e.g. speech chatbot (Week 11, 12)
10. Design multimodal interactions (Week 13, 14)

Text Book

H. SHARP, J. PREECE, Y. ROGERS (2019), Interaction Design: Beyond Human-Computer Interaction, Wiley, 5th Edition.

Course Title	**Maths for Computing**	Course No.	**CSLXXX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-1-0 [4]
Offered for	B.Tech CSE, AIDE	Type	Core
Prerequisite	None		

Objectives

1. To learn about languages, grammars, and computation models
2. To learn about computability
3. To learn basics of parsing techniques

Learning Outcomes:

1. The students are expected to have the ability to:
2. Model computer science problems using discrete mathematical structures
3. Distinguish between computable and incomputable problems
4. Develop understanding of properties of languages and design parsers

Contents

Discrete Structures: Can computers solve every problem? The limits of computing, Set theory, Relations and functions, Propositional logic, First-order logic, Counting techniques. (12 Lectures)
Proof techniques: Constructive, Contraposition, Contradiction; Mathematical Induction (5 Lectures)
Graph Theory: Properties of graphs, Graph matching and coloring. (6 Lectures)
Automata Theory: DFAs, NFAs, Equivalence of DFAs and NFAs, Closure properties of regular languages, Regular expressions, Equivalence of regular expressions and NFAs, Nonregular languages, Context-Free Grammars, Context-Free Languages. (10 Lectures)
Turing Machine: Introduction, Designing Turing machines, The Universal Turing machine. (5 Lectures)
Parsing Techniques: *LR, LALR, Shift-Reduce parsers. (4 Lectures)*

Text Books

M. SIPSER (2014), Introduction to the Theory of Computation, Cengage Learning, 3rd Edition.
K.H. ROSEN (2018), Discrete Mathematics and its Applications, McGraw-Hill, 2018, 8th Edition.

References

J.E. HOPCROFT, R. MOTWANI, J.D. ULLMAN (2008), Pearson, Introduction to Automata Theory, Languages, and Computation, 3rd Edition.
R. JOHNSONBAUGH (2017), Discrete Mathematics, Prentice Hall, 8th Edition.

Self-learning material

Stanford CS103: http://web.stanford.edu/class/cs103/

Course Title	**Design and Analysis of Algorithms**	Course No.	**CSLxxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-1-0 [4]
Offered for	B.Tech CSE, AIDE	Type	Compulsory
Prerequisite	Data Structures and Algorithms		

Objectives

1. To introduce and implement various techniques for designing algorithms and advanced data structures.
2. To learn space and time complexity analysis of algorithms.

Learning Outcomes

1. Ability to choose and implement appropriate algorithm design techniques for solving problems.
2. Understand how the choice of data structures and algorithm design methods impact the performance of programs.
3. Ability to analyze the worst-case and average-case behaviour of algorithms in terms of time and memory requirements.

Contents

Reasoning About Algorithms: P, NP, NP-completeness, Reductions, Complexity analysis. (5 lectures)
Graph Algorithms: Strongly-connected components, Kosaraju's algorithm 1 and 2, Applications. (4 lectures)
Greedy Techniques: Local versus Global optimality, Interval scheduling, Exchange arguments. (5 lectures)
Divide-and-Conquer: Optimality, Recursive algorithms, Divide-and-Conquer recurrences, The Master Theorem and applications, Non-uniform recurrences. (6 lectures)
Dynamic Programming: Reusing sub-computations (Sequence alignment, Bellman-Ford algorithm), Precomputing (Floyd-Warshall algorithm, Johnson's algorithm), Combinatorial problems. (Knapsack) (6 lectures)
Linear Programming: Canonical and standard forms, Feasibility and optimization, Simplex algorithm. (5 lectures)
Approximation Algorithms: Relative approximations, PAS and FPAS scheduling. (4 lectures)
Randomized Algorithms: Random guess (Quick select), Random guess with high confidence (Karger's min-cut algorithm), Storing associative data (Hashing), Error bounds. (7 lectures)

Text Book

T. H. CORMEN, C. E. LEISERSON, R.L. RIVEST, C. STEIN (2009), Introduction to Algorithms, MIT Press, 3rd Edition.

Reference Book

J. KLEINBERG, E. TARDOS (2005), Algorithm Design, Pearson Education, 1st Edition.

Course Title	**Computer Networks**	Course No.	**CSL3xx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-3 [4.5]
Offered for	B.Tech	Type	Compulsory
Prerequisite	Prob. Stat. and Stochastic Process		

Objectives

1. To understand the organization of computer networks, factors influencing the performance of computer networks, and the reasons for having a variety of different types of networks.
2. To understand the Internet structure, various protocols of the Internet and how these protocols address the standard problems of networking and the Internet.
3. Hands-on experience on networking fundamentals through practical sessions.

Learning Outcomes

1. Familiarity with the essential protocols of computer networks in terms of design, implementation & operations.
2. Identifying various design parameters such as latency, bandwidth, error rate, throughput, and their influence on node/link utilization and performance

Contents

Introduction: Layer approach, Packet switching techniques, Performance metrics delay, loss, throughput, bandwidth delay product, latency, Basic understanding of physical and data link layer. (8 Lectures)

Network Layer: Internetworking, Tunneling, Encapsulation, Fragmentation, Internet protocol and its operation, Routing algorithms distance vector and link state algorithm and Routing protocols. (8 Lectures)

Transport Layer: Transmission Control Protocol, Flow control, Error control, Congestion control, Header, Services, Connection management, Timers, Congestion control; User Datagram Protocol, Introduce low latency protocols (WebRTC, LHLS), Applications (IoT and FOG, Real-time applications, ad-hoc wireless protocols). (10 Lectures)

Applications: Network programming, Socket abstraction, Peer-to-peer architecture (P2P architecture), Client server architecture, DNS, HTTP, FTP, SMTP, TelNet, etc. (7 Lectures)

Advanced Internetworking: Multicast routing, Queuing disciplines and buffer management techniques. (6 Lectures)

Network security: Public key and private key cryptography, Digital signature, Firewalls. (3 Lectures)

Laboratory

Networking hardware (Understanding cables, switches, routers, Setting up switching network, Setting up subnets and routing across the subnets)

Socket programming - Development of client-server application using sockets (possible examples, file transfer, peer-peer applications, chat, network monitor etc.)

Networking commands - ifconfig, route, arp, arping, ping, netstat. tcpdump, host, nslookup, dig, ftp, scp, ssh, finger, dhclient, dhcrelay etc.

Protocol analyzer - closely looking at protocols (HTTP, TCP, UDP, ICMP, 802. 3, DHCP, DNS etc.) headers and analyzing the interactions between client and server of different applications

QualNet simulator/Packet Tracer (Implementation of ARQs - Stop-and-wait, Sliding Window Go Back N etc. Verifying operations of routing protocols, influence of congestion on end users performance, congestion control algorithms Reno, New Reno, Cubic, router buffer size on end users performance.

Text Book

K.W. ROSS, J.F. KUROSE (2016), Computer Networks: A Top Down Approach, Pearson, 7th Edition.

Reference Books

W. STALLINGS (2013), Data and Computer Communications, Pearson, 10th Edition.

L.L. PETERSON, B.S. DAVIE (2020), Computer Networks: A Systems Approach, Morgan Kaufmann, 6th Edition.

Course Title	**Database Systems**	Course No.	**CSLXXX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-2 [4]
Offered for	B.Tech	Type	Compulsory
Prerequisite	Data Structures and Algorithms		

Objectives

1. To understand concepts of database design, database management systems and their applications, data modeling, and query languages.
2. To understand different file-structures, transaction management schemes, concurrency control processes, database recovery mechanisms, query processing and optimization.

Learning Outcomes

1. Students are expected to have the ability to:
2. Apply different data modeling methods to requirement analysis, design, and implementation of database systems.
3. Deduce normalized forms for efficient relational database design.
4. Use appropriate storage and access structures.
5. Use techniques for transaction management, concurrency control and recovery.
6. Analyze complexity issues of query execution.

Contents

Preliminaries: Database system concepts and architecture. (2 Lectures)
Semantic Data Model: Entity-Relationship (ER) model, Enhanced Entity-Relationship (EER) model. (4 Lectures)
Relational Model and Languages: Relational data model, Relational database constraints, Relational database design by ER- and EER-to-relational mapping, Relational algebra and Relational calculus. (9 Lectures)
SQL: Schema definition, Constraints, Queries, Views. (5 Lectures)
Database Design Theory: Functional dependencies, Normalization. (5 Lectures)
Algorithms for Query Processing and Query Optimization (5 Lectures)
Database File Organization: Disk storage, File structures, Hashing techniques, Indexing structures. (3 Lectures)
Database Transaction Processing: Transaction concepts and theory, Concurrency control protocols, Database recovery protocols. (6 Lectures)
Advanced Databases: Distributed databases, Object-oriented databases, Multimedia databases, Handling unstructured data, Big data, NoSQL databases. (3 Lectures)

Laboratory

Exercises in SQL covering: Data definition, Data manipulation, Data retrieval, Transaction control
Exercises in PL/SQL covering procedures, functions, loops, triggers
NoSQL data organization and query optimization.

Text Book

R. ELMASRI, S.B. NAVATHE (2017), Fundamentals of Database Systems, Pearson Education, 7th Edition.

Reference Books

1. R. RAMAKRISHNAN, J. GEHRKE (2014), Database Management Systems, McGraw-Hill, 3rd Edition.
2. H.G. MOLINA, J. ULLMAN, J. WIDOM (2014), Database Systems: The Complete Book, Pearson, 2nd Edition.
3. P. RAJ, A. RAMAN, D. NAGARAJ, S. DUGGIRALA (2015), High-Performance Big-Data Analytics: Computing Systems and Approaches, Springer, 1st Edition.
4. N. SABHARWAL, S. GUPTA EDWARD (2014), Big Data NoSQL: Architecting MongoDB, CreateSpace Independent Publishing Platform, 1st Edition.
5. K. BANKER, P. BAKKUM, S. VERCH, D. GARRETT, T. HAWKINS (2016), MongoDB in Action, Manning Publications, 2nd Edition.
6. G. HARRISON (2015), Next Generation Databases: NoSQL and Big Data, Apress, 1st Edition.

Course Title	**Cyber Security**	Course No.	**CSL6XX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3–0–2 [4]
Offered for	B.Tech	Type	Core
Prerequisite	Computer Networks		

Objectives
The Instructor will provide the skills needed to protect networks, secure electronic assets, prevent attacks, ensure the privacy of your customers, and build secure infrastructure.

Learning Outcomes
1. The students are expected to have the ability to:
2. To protect data and respond to threats that occur over the Internet
3. Design and implement risk analysis, security policies, and damage assessment
4. An ability to apply security principles and practices to the environment, hardware, software, and human aspects of a system.

Contents
***Introduction to Cyber Security**:* Internet governance – Challenges and constraints, Cyber threats. (2 Lectures)
Cyber Security Vulnerabilities and Cyber Security Safeguards: Cyber security, Vulnerabilities, safeguards, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical hacking, Firewalls, Response, Scanning, Security policy, Threat management. (8 Lectures)
Securing Web Application, Services and Servers: Basic security for HTTP applications and services, Basic security for SOAP services, Identity management and Web services, Authorization patterns, Security considerations, challenges. (8 Lectures)
Intrusion Detection and Prevention: Intrusion detection and Prevention techniques, Anti-malware software, Security information management, Network session analysis, System integrity validation. (7 Lectures)
Overview of Firewalls: Types of firewalls, User management, VPN security protocols: - PGP and S/MIME, Security at transport layer- SSL and TLS, Security at network layer-IPSec. (5 Lectures)
Cyberspace and the Law: Cyber security regulations, Roles of international law, State and Private Sector in cyberspace, Cyber security standards. The INDIAN cyberspace, National cyber security policy. (6 Lectures)
Cyber Forensics: Handling preliminary investigations, Controlling an investigation, Conducting disk-based analysis, Investigating information-hiding, Tracing internet access, Tracing memory in real-time. (6 Lectures)

Laboratory
Design and implementation of a simple client/server model and running application using sockets and TCP/IP.
To make students aware of the insecurity of default passwords, printed passwords and passwords transmitted in plain text.
To teach students how to use SSH for secure file transfer or for accessing local computers using port forwarding technique.
Comparison between Telnet and SSH for Secure Connection
AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications

Text Book
C.J. HOOFNAGLE (2016), Federal Trade Commission Privacy Law and Policy, Cambridge University Press, 2016.

Self Learning Material
P.W. SINGER, A. FRIEDMAN (2014), Cybersecurity: What Everyone Needs to Know, OUP, 1st Edition.
L. THAMES, D. SCHAEFER (2017), Cybersecurity for Industry 4.0, Springer, 1st Edition.
N. HASSAN, R. HIJAZI (2017), Digital Privacy and Security Using Windows, Apress, 1st Edition.

Course Title	**Computer Architecture**	Course No.	**CSLXXX/EELXXX**
Department	CSE and EE	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B. Tech. CSE	Type	Compulsory
Prerequisite	None		

Objectives

1. To understand aspects of computer architecture and program performance
2. To provide essential understanding of different subsystems of modern computer system and design aspects these subsystems
3. To introduce hands-on experience of computer architecture design and performance enhancement

Learning Outcomes

1. The students are expected to have the ability to:
2. Ability to identify the basic components and design of a computer, including CPU, memories, and input/output units
3. Ability to identify the issues involved in the instruction execution and various stages of instruction life stage
4. Ability to identify the issues related to performance improvement

Contents

Introduction: Basic computer organization, Components of computer systems, information representation. (3 Lectures)

Central Processing Unit: Arithmetic and Logic Unit; Instruction sets; RISC, CISC, and ASIC/ASIP paradigms; Various addressing modes; Assembly language programming; Instruction interpretation: micro-operations and their RTL specification; CPU design, Hardwired and microprogrammed, Performance issues: Parallel processing, Pipelining, Hazards, Advanced parallelization techniques. Cache Coherence protocols, Multicore Architecture (16 Lectures)

Memory Hierarchy: Memory organization, Various levels of memory architecture and their working principles, Cache memory, Writing strategy, Coherence, Performance issues and enhancement techniques for memory design. (14 Lectures)

Interfacing: I/O transfer techniques: Program controlled, Interrupt controlled and DMA; Introduction to computer buses, Peripherals and current trends in architecture. (9 Lectures)

Text Books

D.A. PATTERSON, J.L. HENNESSY (2008), Computer Organization and Design, Morgan Kaufmann, 4th Edition.

W. STALLINGS (2015), Computer Organization and Architecture: Designing for Performance, Pearson Education India, 10th Edition.

References

A.S. TANENBAUM (2013), Structured Computer Organization, Prentice Hall of India, 6th Edition.

V.C. HAMACHER, Z.G. VRANESIC, S.G. ZAKY (2011), Computer Organization, McGraw Hill, 5th Edition.

J.L. HENNESSY, D.A. PATTERSON (2017), Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 6th Edition.

D.V. HALL, S.S.S.P. RAO (2017), Microprocessors and Interfacing, McGraw Hall, 3rd Edition.

Course Title	**Software Engineering**	Course No.	**CSL2XX**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-2 [4]
Offered for	B.Tech CSE	Type	Compulsory
Prerequisite	Introduction to Computer Science		

Objectives

1. This course focuses on engineering practices and processes that development team uses to make sure the team is setup for change
2. The course introduces practices, techniques and processes that can help team build high quality software

Learning Outcomes

Able to comfortably and effectively participate in various techniques and processes for building scalable and high quality software.

Contents

Introduction of Software Engineering: Need for software engineering, Software quality attributes, Software product pipelines, Software life cycle models and processes, Requirement engineering using UML Diagrams. (9 Lectures)

Software Architecture and Design: Design principles, Design Patterns, Architecture Versus Design, Modularity, Software Components and Connectors, Architecture Styles. (6 Lectures)

Essence of Modern Software Engineering: Software engineering essence, Essence language, Essence kernel, Using essence kernel in agile development practices, Agile Principles, Agile process models through essence kernel, Large scale complex development Using kernel. (13 Lectures)

Software Testing: Quality metrics, Coding style and Static analysis tools, Verification and validation, Various testing techniques and Test case generations. (7 Lectures)

Software Project Management: Software versioning and Continuous integration, Project management and Risk analysis, Configuration management, Cost analysis and estimation. (7 Lectures)

Laboratory

Assignment/Project on Software requirement acquisition, UML diagrams, Preparing software requirement specification, Practicing agile methods, User story, Backlog, Test case generation, Unit testing, CI configurations, Cost estimation, Manpower management and Sprint analysis using Burn down charts.

Text Books

R.S. PRESSMAN, B.R. MAXIM (2019), Software Engineering: A Practitioner's Approach, McGraw-Hill India, 2019, 9th Edition.

Mark Richards, Neal Ford (2020), Fundamentals of Software Architecture, O'Reilly Media, Inc.

L. BASS, P. CLEMENTS, R. KAZMAN (2012), Software Architecture in Practice, Pearson, 3rd Edition.

I. JACOBSON, H. LAWSON, P.W. NG, P.E. McMAHON, M. GOEDICKE (2019), The Essentials of Modern Software Engineering, ACM Books.

Course Title	**Principles of Programming Languages**	Course No.	**CSXXX**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech	Type	compulsory
Prerequisites	Data Structures and Algorithms, Computer Architectures, Operating Systems		

Objectives

1. To introduce major programming paradigms, and principles and techniques involved in design and implementation of modern programming languages.
2. To introduce frameworks for reasoning about programming languages.

Learning Outcomes

1. To identify the abstract syntax of any programming language
2. To be able to analyze and critique/appreciate the inherent philosophy of different programming languages
3. To be able to design small domain-specific languages/language features and implement them as interpreters or embeddings in another language.

Contents

Introduction: Study of principles of computing paradigms like imperative, functional, object-oriented and logic programming. (2 Lectures)
Syntax and semantics of programming languages: Syntax and semi-formal semantic specification using attribute grammar. (4 Lectures)
Imperative Programming: Location, Reference and expressions, Assignment and Control, Data types, Blocks, Procedures, Modules. (6 Lectures)
Object-oriented programming: Classes, Objects, Abstraction, Encapsulation, Inheritance and software reuse. (6 Lectures)
Functional programming: Function as first-class objects, Higher order functions, Polymorphic data types, Type checking, Type inferencing, Recursion, Lambda calculus. (6 Lectures)
Logic programming: Horn clauses, Unification, SLD resolution, Backtracking, Cuts. (6 Lectures)
Concurrent programming: Processes, Synchronization primitives, Safety and liveness properties, Multithreaded programs. (6 Lectures)
Case Studies: C++, Java, Haskell, Prolog, Python. (6 Lectures)

Text Books

R. SETHI (2006), Programming Languages: Concepts and Principles, Pearson, 2nd Edition.
T. W. PRATT, M.V. SELKOWITZ (2000), Programming Languages: Design and Implementation, Pearson, 4th Edition.

Reference Books

B.J. MacLENNAN (1999), Principles of Programming Languages: Design, Evaluation and Implementation, Oxford University Press, 3rd Edition.
M. BEN-ARI (2005), Principles of Concurrent and Distributed Programming, Pearson, 2nd Edition.
J.C. MITCHELL (1996), Foundations for Programming Languages, MIT Press.
B.C. PIERCE (2002), Types and Programming Languages, MIT Press, 1st Edition.
H.P. BARENDREGT (2014), The Lambda Calculus: Its Syntax and Semantics, North-Holland, 2nd Edition.
D.P. FRIEDMAN, M. WAND (2008), Essentials of Programming Languages, MIT Press, 3rd Edition.

15. Detailed Course Content of Programme Elective Courses

Core Bouquet - 1

1. Randomized Algorithms 7xx
2. Approximation Algorithms 7xx
3. Distributed Algorithms 7xx
4. Graph Theoretic Algorithms 7xx
5. Parameterized Complexity 7xx
6. Optimization (Mathematics) 4xx

Course Title	**Randomized Algorithms**	Course No.	**CS7xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE, AIDE	Type	Bouquet/Elective
Prerequisite	Design and Analysis of Algorithms, Probability, Statistics and Stochastic Processes		

Objectives

This course presents basic concepts in the design and analysis of randomized algorithms.

Learning Outcomes

Familiarity with several of the main thrusts of work in randomized algorithms.
Ability to read current research publications in the area.
Ability to use randomized algorithm techniques for real world problems.

Contents

Tools and Techniques: Basic probability theory; Randomized complexity classes; Game-theoretic techniques; Markov, Chebyshev, and moment inequalities; Limited independence; Tail inequalities and the Chernoff bound; Conditional expectation; The probabilistic method; Markov chains and random walks; Algebraic techniques; Probability amplification and derandomization. (22 Lectures)
Applications: Sorting and searching; Data structures; Combinatorial optimization and graph algorithms; Geometric algorithms and linear programming; Approximation and counting problems; Parallel and distributed algorithms; Online algorithms. (20 Lectures)

Text Books

R. MOTWANI, P. RAGHAVAN (1995), Randomized Algorithms, Cambridge University Press, 1st Edition.

Reference Books

M. MITZENMACHER, E. UPFAL (2017), Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2nd Edition.
W. FELLER (2008), An Introduction to Probability Theory and Its Applications, Volumes I and II, John Wiley, 2nd Edition.
P. BILLINGSLEY (2012), Probability and Measure, John Wiley.

Self Learning

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-856j-randomized-algorithms-fall-2002/lecture-notes/.

Course Title	**Approximation Algorithms**	Course No.	**CS7xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE, AIDE	Type	Bouquet/Elective
Prerequisite	Design and Analysis of Algorithms		

Objectives

The objective of the course is to introduce general-purpose techniques for large classes of intractable optimization problems.

Learning Outcomes

Students will be able to:

1. Identify novel and significant open research questions in the field.
2. Apply their knowledge in the design and implementation of viable solutions to those problems.

Contents

Combinatorial Techniques: Metric TSP, Euclidean TSP, FPTAS for knapsack, Greedy algorithms for makespan, PTAS for makespan, Local search: The Min degree spanning tree problem. (10 Lectures)
Linear Programming Relaxations: Linear programming: Vertex cover, Set cover via Dual fitting, Set cover via the Primal-Dual schema, Steiner forest, Facility location. (10 Lectures)
Cuts, Metrical Relaxations, and Embeddings: Basic embedding results and applications, Tree embeddings: Buy at bulk network design and L1 embeddings, Min multicut and metrical relaxation for sparsest cut, Rounding for sparsest cut via L1 embeddings, Balanced cuts and their applications, Tree embeddings via random cuts. (10 Lectures)
Random Hyperplanes, Dimensionality Reduction, and Hashing: Solving semidefinite programs, Rounding SDPs using random projections: MAX-CUT, Dimensionality reduction via random projections, Approximate nearest neighbors, Locality sensitive hashing: Random projections and Min hashing. (10 Lectures)
Game Theory: Selfish algorithmic strategies, Cost sharing schemes using Primal-Dual algorithms. (2 Lectures)

Text book

V. VAZIRANI (2013), Approximation Algorithms, Springer-Verlag.

Reference Books

1. M.R. GAREY, D.S. JOHNSON (2016), Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman Company.
2. D. DU, K.KO, X. HU (2011), Design and Analysis of Approximation Algorithms, Springer.

Course Title	**Distributed Algorithms**	Course No.	**CS7xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE, AIDE	Type	Bouquet/Elective
Prerequisite	Design and Analysis of Algorithms		

Objectives
The objective of the course is to introduce the distributed algorithms in real-world systems

Learning Outcomes
Students will be able to:
1. To introduce the most important basic results in the area of distributed algorithms, and prepare interested students to begin independent research or take a more advanced course in distributed algorithms.
2. To implement distributed algorithms in real-world distributed computing platforms
3. To understand and identify applications of distributed algorithms in real-world systems

Contents
Introduction to Distributed Computing Models : Model of a distributed system: asynchronous message-passing model, Time and message complexity, Safety and liveness properties, Clock Synchronization [5 Lectures]
Leader Election Logical Clocks and Causality: *Leader election in rings, Asynchronous leader election with identities, Synchronous leader election by abusing the synchronous model, Logical clocks and vector clocks , Snapshots and synchronization [7 Lectures]*
Message Ordering and Group Communication: Termination Detection Algorithms and Reasoning with Knowledge [4 Lectures]
Distributed Mutual Exclusion Algorithms: *Lamport's algorithm, Ricart–Agrawala algorithm, and Deadlock Detection Algorithms* [6 Lectures]
Global Predicate Detection: Modalities on predicates, Centralized algorithm for relational predicates, Conjunctive predicates, Distributed algorithms for conjunctive predicates [3 Lectures]
Distributed Shared Memory: Memory consistency models, Shared memory mutual exclusion and Wait-freedom [6 Lectures]
Checkpointing and Rollback Recovery: Checkpoint-based recovery and Log-based rollback recovery [3 Lectures]
Consensus and Agreement : Agreement in a failure-free system (synchronous or asynchronous) , Agreement in (message-passing) synchronous systems with Failures, Agreement in asynchronous message-passing systems with failures , Wait-free shared memory consensus in asynchronous systems [6 Lectures]
Distributed Programming Environments: Communication primitives, selected case studies [2 Lectures]

Text book
A.D. KSHEMKALYANI, M. SINGHAL (2011), Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press.

Reference Books
1. N. LYNCH, (1996), Distributed Algorithms, Morgan Kaufmann, 1st Edition.
2. G. TEL, (2000), Introduction to Distributed Algorithms, Cambridge University Press, 2nd Edition.

Title	**Graph-theoretic Algorithms**	Number	**CS7xxx**
Department	Computer Science	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech., M. Tech.	Type	Elective
Prerequisite	Algorithm Design and Analysis		

Objectives

The Instructor will:

Introduce and address graph problems using algorithm design principles.

Discuss graph theoretic algorithms in real-world scenarios.

Learning Outcomes

To give the student further exposure to the design, analysis, and application of algorithms for problems defined on graphs.

Formulate and solve real-world problems using the mathematical foundations of graph theory.

Contents

Introduction to graphs. (2 Lectures)

Max Flow Min Cut Theorem, Algorithms for min cut and max flow, Maximum matching, Hall's theorem, algorithms for computing maximum matching in weighted and unweighted graphs. (6 Lectures)

Vertex and Edge Coloring of Graphs, Independent Set. (4 Lectures)

Interval graphs, Perfect elimination orders, Comparability graphs, Interval Graph Recognition, Friends of Interval Graphs. (7 Lectures)

Trees and treewidth. (7 Lectures)

Planar Graphs, Problems in planar graphs, Maximum Cut. (8 Lectures)

Planarity Testing, Triangulated graphs, Friends of planar graphs, Hereditary properties. (8 Lectures)

Textbooks

A. BRANDSTÄDT, V.B. LE, J.P. SPINRAD (1999), Graph Classes: A Survey, SIAM Monographs on Discrete Mathematics and Applications, SIAM.

T. NISHIZEKI, N. CHIBA (1988), Planar Graphs: Theory and Algorithms, North-Holland publisher.

M.C. GOLUMBIC (2004), Algorithmic Graph Theory and Perfect Graphs, Academic Press, 2nd Edition.

Self-Learning Material

T. BIEDL, Lecture Notes of a Graduate Course, University of Waterloo.

Title	**Parameterized Complexity**	Number	**CS7xxx**
Department	Computer Science	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech., M. Tech.	Type	Elective
Prerequisite	Algorithm Design and Analysis		

Objectives

The Instructor will:

The objective of the course is to introduce a technique to deal with NP-hard problems

Learning Outcomes

Learn a new set of techniques to cope with NP-hard problems.

Identify novel and significant open research questions in the field.

Contents

Introduction to Parameterized Complexity. (1 Lecture)

Basic Techniques to design parameterized algorithms: Bounded Search Tree, Iterative Compression, Randomized method. (16 Lectures)

rnelization: Greedy Based Kernels, Matching Based Kernels, Sunflower Lemma, Crown Decomposition, Expansion Lemma. (16 Lectures)

Parameterized Intractability. (9 Lectures)

Textbooks

M. CYGAN, F. FOMIN, L. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, S. SAURABH (2015), Parameterized Algorithms, Springer.

F. FOMIN, D. LOKSHTANOV, S. SAURABH, M. ZEHAVI (2019), Kernelization: Theory of Parameterized Preprocessing, Cambridge University Press.

Self-Learning Material

https://www.youtube.com/watch?v=Ex8TueBsF1g&list=PLhkiT_RYTEU0gpi97fqjtaHy9Gk47oF85&index=1

https://www.youtube.com/watch?v=VAEul4J3Br8&list=PLhkiT_RYTEU1ekkuDwau01N4_Axr_PHQI&index=1

Core Bouquet - 2

1. VLSI Design (EE) 3xx
2. Embedded Systems (EE) 3xx
3. High-level Synthesis 7xx

Course Title	**High-level Synthesis**	Course No.	**CS/EE-7XX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	M.Tech	Type	Elective
Prerequisite	Basic knowledge of digital circuits		

Objectives

To provide an introduction to high-level synthesis of digital circuits and electronic design automation.
To explain the HLS design flow: circuit, logic, RTL, algorithm, systems.

Learning Outcomes

The students are expected to have the ability in:
Design Digital Circuits, Logic synthesis, optimization and technology mapping.
Design Verification
Use different tools like C-to-Silicon, AutoPilot from AutoESL, Synopsis for logic synthesis,

Contents

Introduction: Basic overview of the VLSI design flow, design abstractions (2 Lectures)
Digital Sub-System Design: Logic Circuits, Adders, Multipliers, ALU (5 Lectures)
HDL language: System C (8 Lectures)
Synthesis: Scheduling, Allocation, Binding (3 Lectures)
RTL Synthesis, Retiming, FSM Encoding, Combinational logic optimization and technology mapping (6 Lectures)
Verification: Design vs verification, Types of Verification at each step of ASIC Design Flow (functional/power/Gate Level Simulation (GLS)), UVM, Verification planning and Designing a test bench (7 Lectures)
Chip Architectures: Full Custom, Standard Cells and FPGA (4 Lectures)
Synthesis Tool: C-to-Silicon, AutoPilot from AutoESL (4 Lectures)
Overview of layout synthesis topics: Placement and routing. Chip Synthesis. (3 Lectures)

Reference Books

M. FINGEROFF (2010), High-level Synthesis Blue Book, Xlibris.
G. De MICHELI (2017), Synthesis and Optimization of Digital Circuits, McGraw-Hill.
S. KANG, Y. LEBLEBICI (2003), MOS Digital Integrated Circuits, Analysis and Design, WCB-MCGrawHill, 3rd Edition.

Theory and Algorithms

1. Theory of Computation 4xx
2. Randomized Algorithms 7xx
3. Approximation Algorithms 7xx
4. Complexity Theory 4xx
5. Advanced Data Structures 4xx
6. Stream Analytics
7. Quantum Computing (IDRP) 4xx
8. Optimization (Mathematics) 3xx

Title	**Theory of Computation**	Course No.	CSL4xxx
Department	Computer Science and Engineering	Structure (L-T-P [C])	
Offered for	B.Tech (CSE)	Type	Elective
Prerequisite	Mathematics for Computing	Antirequisite	

Objectives

To learn about computability techniques
To learn about advanced computational complexity models

Learning Outcomes

To be able to distinguish between computable and un-computable problems
Gaining in-depth understanding of advanced complexity models

Contents

Computability: Review of Turing Machine, NP and NP-completeness, Diaganolization, view of PDAs, 2DFAs, FAs as restricted TMs, and related theorems. Tape reduction, and robustness of the model. Encoding and Enumeration of Turing Machines, Undecidability. Rice-Myhill-Shapiro theorem. Relativisation.

Decision Trees and Communication Complexity:
Certificate Complexity, Randomized Decision Trees, Lower bounds on Randomized Complexity, Some techniques for decision tree lower bounds, Comparison trees, and sorting lower bounds, Yao's MinMax Lemma, Definition of communication complexity, Lower bound methods, Overview of other communication models, Applications of communication complexity.

Time Complexity: Time as a resource, Linear Speedup theorem, Crossing sequences and their applications, Hierarchy theorems. P vs NP. Time Complexity classes and their relationships. Notion of completeness, reductions. Cook-Levin Theorem. Ladner's theorem. Relativization Barrier: Baker-Gill-Solovoy theorem.

Space Complexity: Space as a resource. PSPACE, L and NL. Reachability Problem, Completeness results. Savitch's theorem, Inductive Counting to show Immerman-Szelepscenyi theorem. Reachability problems, Expander Graphs, SL=L

Complexity of Counting & Randomization: Counting problems, Theory of #P-completeness. The complexity classes PP, ParityP, BPP, RP, BPP is in P/poly, Toda's theorem.

Textbooks

D. Kozen (2013), Automata and Computability, Springer
D. KOzen (2006), Theory of Computation, Springer
S. Arora and B. Barak (2009), Complexity Theory: A Modern Approach, Cambridge University Press.

Reference books

Sipser, M., (2013), Introduction to the Theory of Computation, Cengage Learning
Hopcroft,J. E., Motwani, R., and Ullman, J. D., (2007), Introduction to Automata Theory, Languages, and Computation, Pearson

Self-Learning Material

Course Title	**Complexity Theory**	Course No.	**CSL6xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE	Type	Elective
Prerequisite	Maths for Computing, Data Structures and Algorithms		

Objectives

The course provides a challenging introduction to some of the central ideas of theoretical computer science. It attempts to present a vision of computer science beyond computers.

Learning Outcomes

Students are expected to understand:

The theory of classical complexity classes and its challenges.

The state-of-the-art in the field of complexity theory.

Contents

Basic Complexity Classes: The computational model, NP and NP completeness, Diagonalization, Space complexity, The Polynomial hierarchy and alternations, Circuits, Randomized computation, Interactive proofs, Complexity of counting, Complexity issues in cryptography. (16 Lectures)

Lower Bounds for Concrete Computational Models: Decision trees, Communication complexity, Circuit lower bounds, Algebraic computation models. (10 Lectures)

Advanced topics: Average case complexity: Levin's theory, Derandomization, Expanders and extractors, Hardness amplification and Error correcting codes, PCP and hardness of approximation, Logic in complexity theory, Quantum computation. (16 Lectures)

Text Books

S. ARORA, B. BARAK (2009), Computational Complexity: A Modern Approach, Cambridge University Press, 1st Edition.

Self Learning

MIT Lecture Notes, Automata, Computability, and Complexity, Electrical Engineering and Computer Science: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/lecture-notes/

Course Title	**Advanced Data Structures**	Course No.	**CS4xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE, AIDE	Type	Elective
Prerequisite	Data Structures and Algorithms		

Objectives

Data structures are one of the important elements of modern computer science. The objective of the course is to learn major results and current research directions in data structure beyond the traditional linear and non-linear data structures.

Learning Outcomes

Students will be able to use unconventional and modern data structures for the changing need of new age algorithmic problems.

Students will learn to solve problems using data structures.

Contents

Temporal Data Structures: Persistent and retroactive data structures, Pointer machine. (5 Lectures)

Geometric Data Structures: Point location, Range searching, Fractional cascading, Data structures for moving data. (5 Lectures)

Dynamic optimality: Analytic bounds, Splay trees, Geometric view, independent rectangle, key-independent optimality. (5 Lectures)

External Memory Data Structures: Cache-oblivious (CO) Data structures such as B-Tree, CO priority Queue etc. (5 Lectures)

Integer Data Structures*:* Predecessor/Successor problems, van Emde Boas, x-fast and y-fast trees, Fusion trees: Sketching, Parallel comparison, Most significant set bit. (6 Lectures)

Static Trees and String Data Structures: Least common ancestor, Range minimum queries, Level ancestor, Suffix tree, Suffix array, Linear-time construction for large alphabets, Suffix tray, Document retrieval. (5 Lectures)

Memory Efficient Data Structures*:* Rank, Select, Tries, Compact suffix arrays and trees. (5 Lectures)

Dynamic Graphs: Link-cut trees, Heavy-light decomposition, Euler tour trees, Decremental connectivity in trees, Fully dynamic connectivity. (6 Lectures)

Reference Books

R.E. TARJAN (1983), Data Structures and Network Algorithms, SIAM. (covers BSTs, splay trees, link-cut trees)

P. MORIN (2013), Open Data Structures, Athabasca University Press, 31st Edition. (covers BSTs, B-trees, hashing, and some integer data structures)

Self Learning Material

MIT Lectures in Advanced Data Structures (6.851): https://courses.csail.mit.edu/6.851/fall17/lectures/

Course Title	**Stream Analytics**	Course No.	**CSxxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech 3rd and 4th years; PG	Type	Elective
Prerequisite	Introduction to Machine Learning, Pattern Recognition and Machine Learning		

Objectives

Provide background on some of the important models, algorithms, and applications related to stream data.

Learning Outcomes

Ability to understand and apply the practical and algorithmic aspects related to various topics of data streams

Contents

Introduction: Stream and mining algorithms. (2 Lectures)

Clustering Massive Data Streams: Micro-clustering based stream mining, Clustering evolving data streams, Online micro-cluster maintenance, High-dimensional projected stream clustering, Classification of data streams using micro-clustering, On-demand stream classification, Applications of micro-clustering. (12 Lectures)

Classification Methods in Data Streams: Ensemble based classification, Very fast decision trees, On-demand classification, Online Information Network. (6 Lectures)

Distributed Mining of Data Streams: Outlier and anomaly detection, Clustering, Frequent itemset mining, Classification, Summarization. (6 Lectures)

Change Diagnosis Algorithms in Evolving Data Streams: Velocity density method, Clustering for characterizing stream evolution. (4 Lectures)

Multidimensional Analysis of Data Streams using Stream Cubes: Architecture for online analysis of data streams, Stream data cube computation, Performance study. (6 Lectures)

Dimensionality Reduction and Forecasting on Streams: Principal Component Analysis, Auto-regressive models and recursive least squares, Tracking correlations and hidden variables. (6 Lectures)

Text Book

C.C. AGGARWAL, (Ed.), Data Stream: Models and Algorithms, Kluwer Academic Publishers, 2007.

Hardware

1. Embedded Systems (EE) 3xx
2. VLSI Design (EE) 3xx
3. High-level Synthesis 4xx
4. Hardware Software Co-Design 7xx
5. Neuromorphic Computing and Design 7xx

Title	**Hardware Software Co-design**	Number	EEL7XX/CS7XX
Department	CSE/Electrical Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech.	Type	Elective
Prerequisite	Digital Design		

Objectives

The Instructor will:

Give overview of the cooperative design of hardware and software components.

Provide the understanding of unification of currently separate hardware and software paths

Learning Outcomes

The students are expected to have the ability to:

Achieve system-level objectives by exploiting the synergism of hardware and software through their concurrent design.

Understand the movement of functionality between hardware and software.

Contents

Introduction to Hardware-Software Co-design, Design process, Application domains, Design technologies. (5 Lectures)

Data flow modeling implementation in software and hardware, Analysis of control flow and data flow, Finite state machine with datapath, Custom hardware, Microprogrammed architecture, system on chip. (12 Lectures)

Codesign methodologies: Model representation, Hardware-software mapping - partitioning, scheduling, and allocation, Software code optimization. (10 Lectures)

Hardware-software interfaces: Principles of hardware software communication, On-chip buses, Memory-mapped interface, Coprocessor interface. (10 Lectures)

Codesign CAD tools: Ptolemy Project. (5 Lectures)

Textbook

P.R. SCHAUMONT (2010), A Practical Introduction to Hardware/Software Codesign, Springer; 1st Edition.

G. De MICHELI, M. SAMI (1996), Hardware/Software Co-Design, Springer.

Course Title	**Neuromorphic Computing and Design**	Course No.	**CS7xx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech, MTech, PhD	Type	Elective
Prerequisite	Introduction to Machine Learning / Pattern Recognition and Machine Learning		

Objectives

To provide information about neuroscientific progress towards reverse-engineering the brain

To provide essentials on key hardware building blocks, system level VLSI design and practical real-world applications of neuromorphic Systems

Learning Outcomes

The students are expected to have the ability to:

View neuromorphic computing as a computer architecture research problem

Perform software and hardware implementation of basic biological neural circuits

Contents

(*Fractal 1*)

Foundational Concepts: Introduction to neuromorphic engineering, Neuroanatomy of human brain, Signaling and operation of biological neurons, Neuron models - LIF, IF, HH, Synapses and plasticity rules, Spike-time-dependent plasticity (STDP), Biological neural circuits, Non-von Neumann computing approach, Learning rules, Retina, Cochlea. (14 Lectures)

(*Fractal 2*)

Neuromorphic Computing: Spiking Neural Networks (SNN), Advanced nanodevices for neuron implementation, Synaptic emulation - Non-volatile memory (NVM), Flash, RRAM, Memristors, CNT, Case study on Intel's Loihi neuromorphic chip. (14 Lectures)

(*Fractal 3*)

Hardware Implementation: Electronic synapses, Digital/Analog neuromorphic VLSI, Hardware implementation of neuron circuits, Hardware implementation of synaptic and learning circuits, Synaptic programming methodology optimization. (14 Lectures)

Text Books

S. C. LIU (2002), Analog VLSI: Circuits and Principles, MIT Press.

R. KOZMA (2012), Advances in Neuromorphic Memristor Science, Springer.

E. KANDEL (2012), Principles of Neural Science, McGraw Hill.

Systems

1. Software Testing and Quality Assurance 7xx
2. Distributed Database Systems 7xx
3. Distributed Systems 4xx
4. Mobile and Pervasive Computing 700
5. Edge and Fog Computing 4xx
6. Virtualization and Cloud Computing 700
7. Compiler Design 7xx
8. Software Maintenance 4xx

Course Title	**Software Testing and Quality Assurance**	Course No.	**CS7xxx**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech CSE	Type	Elective
Prerequisite	Software Engineering		

Objectives

The course provides an introduction to software engineering testing process, describes the quality assurance process and its role in software engineering.

Learning Outcomes

The student will be able to work with various testing techniques, methods and tools.
The student will be able to demonstrate proficiency in managing a industry scale software project to customer requirements.
Students will be able to describe the state of the art validation and verification techniques.

Contents

Overview and Basics: Software quality, Quality assurance and its context, Quality Engineering. (6 Lectures)
Software Testing: Concepts, Issues, and Techniques, Testing process, Testing levels such as unit, module, subsystem, system; Automatic and Manual techniques for generating and validating test data, Static vs dynamic analysis, Functional testing, Web application testing, Reliability assessment. (20 Lectures)
Quality Assurance beyond testing: Defect prevention and Process improvement, Software inspection, Formal verification, Fault tolerance and Failure containment, Comparing quality assurance techniques and activities. (8 Lectures)
Quantifiable Quality Improvement: Feedback loop and activities for Quantifiable quality improvement, Quality model and measurements, Defect classification and analysis, Risk identification, Reliability engineering. (8 Lectures)

Reference Books

J. TIAN (2005), Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement, Wiley.
P. AMMANN, J. OFFUTT (2016), Introduction to Software Testing, Cambridge University Press, 2nd Edition.

Course Title	**Distributed Database Systems**	Course No.	**CS7XX**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech, MTech, PhD	Type	Elective
Prerequisites	Operating Systems, Database Systems, Computer Networks, Data Communication		

Objective

To understand and appreciate concepts of distributed database design, and its associated issues of consistency, concurrency, optimization, integrity, reliability, privacy, and security.

Learning Outcome

Ability to understand the need for distributed database systems and its related complexities pertaining to fragmentation, replication, availability, concurrency, consistency and recovery.

Contents

Introduction: Distributed data processing concepts, What is a DDBS - advantages, disadvantages and problem areas. (2 Lectures)
Distributed Database Management System Architectures: Transparencies, Architecture, Global directory concepts and issues. (3 Lectures)
Distributed Database Design: Design strategies, Design issues, Fragmentation, Data allocation. (4 Lectures)
Semantics Data Control: View management, Data security, Semantic integrity control. (5 Lectures)
Query Processing: Objectives, Characterization of processors, Layers of processing, Query decomposition, Data localization. (5 Lectures)
Query Optimization: Factors, Centralized query optimization, Fragmented query ordering, Query optimization algorithms. (5 Lectures)
Transaction Management: Goals, Properties, Models. (4 Lectures)
Concurrency Control: Concurrency control in centralized systems, Concurrency control in DDBSs - algorithms, Deadlock management. (5 Lectures)
Reliability: Issues and types of failures, Reliability techniques, Commit protocols, Recovery protocols. (5 Lectures)
Other Avenues: Parallel database Systems, Multi-databases. (4 Lectures)

Reference Books

S. CERI, G. PELAGATTI (2008), Distributed Databases: Principles and Systems, McGraw-Hill, 1st Edition (2017 Reprint).
M.T. ÖZSU, P. VALDURIEZ (2011), Principles of Distributed Database Systems, Springer, 3rd Edition.

Course Title	**Distributed Systems**	Course No.	**CS4XX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	2-0-0 [2]
Offered for	B.Tech	Type	Elective
Prerequisite	Operating Systems		

Objectives

To introduce issues, challenges and approaches related to distributed systems

Learning Outcomes

The students are expected to have the ability in:
Designing, deploying and managing distributed systems
Understanding and analyzing various distributed algorithms and approaches

Contents

Introduction: Introduction to advanced operating systems and distributed systems. (2 Lectures)
Concurrency, Ordering, Races: Implementing Remote Procedure Calls, Time, Clocks, and the Ordering of events in a distributed system, Distributed snapshots: Determining global states of distributed systems, Detecting concurrency bugs: Eraser & TSVD. (8 Lectures)
File Systems and Disks: A Fast file System for UNIX, Scale and performance in a distributed file system, The Design and implementation of a log-structured file system, A case for redundant arrays of inexpensive disks (RAID). (6 Lectures)
Fault Tolerance: Implementing fault-tolerant services using the state machine approach, Paxos, Practical Byzantine fault tolerance. (6 Lectures)
OS Kernels and Virtual Machines: Microkernels, System code verification, Kernels for multicore & disaggregation, Virtual machines. (6 Lectures)

Text Books

T. ANDERSON, M.DAHLIN (2014), Operating Systems: Principles and Practice, Recursive Books, 2nd Edition.
N. LYNCH (2009), Distributed Algorithms, Elsevier India.

Reference Books

W.R. STEVENS, S.A.RAGO (2013), Advanced Programming in the Unix Environment, Addison-Wesley, 3rd Edition.
W.R. STEVENS, B. FENNER, A.M. RUDOFF (2004), Unix Network Programming: Networking APIs: Sockets and XTI (Volume 1), Addison-Wesley, 3rd Edition.

Online References

http://pages.cs.wisc.edu/~remzi/Classes/537/Spring2018/
Https://www.cs.cmu.edu/~15712/syllabus.html

Course Title	**Mobile and Pervasive Computing**	Course No.	**CS7XX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B. Tech and M. Tech	Type	Elective
Pre-requisites	Computer Networks		

Objectives

1. Explain necessary concepts to understand challenges in developing applications for mobile devices, when contrasted with that of a general-purpose computer.
2. Introduce mechanisms employed by different cellular technologies, particularly 5G, and contrast them with each other.

Learning Outcomes

At the end of this course, students would be able to:

1. Describe the working principles of various cellular technologies, as well as appreciate the associated challenges.
2. Compare/contrast various aspects of Mobile and Pervasive Computing, viz. localization, sensing, security and privacy concerns, application development, etc.
3. Implement the theoretical concepts into practical scenarios..

Contents

Overview of Wireless Systems: Infrastructure-based vs Ad-hoc, Wireless LANs, Cellular systems, Sensor networks, Bluetooth, WiFi, WiMAX. (3 Lectures)
Medium Access: Link Adaptation, Routing Protocols. (4 Lectures)
Mobility and Handoff Management: Link layer mobility mechanisms (location management protocols), Network layer mobility mechanisms (Macro and Micro mobility protocols), Handoff management protocols. (6 Lectures)
Cellular Networks: LTE and 5G overview, 5G Architecture, RAN and dynamic CRAN, Mobility management and Network slicing in 5G. (11 Lectures)
Pervasive Computing: Principles, Characteristics, Pervasive devices, Smart sensors and actuators, Context communication and access services. (4 lectures)
Context Aware Sensor Networks: Open protocols (SDP, Jini, SLP, UPnP), SyncML framework, Context aware mobile services, Context aware security. (4 lectures)
Energy Efficiency, Localization **(GPS/WiFi/GSM),** ***Security*** (4 Lectures)
Recent Advances in Wearable devices (3 Lectures)
Body Area Networks **(BAN)** (3 Lectures)

Textbooks

1. I. STOJMENOVIC (2002), Handbook of Wireless Networks and Mobile and Pervasive Computing, Wiley.
2. S. LOKE (2006), Context-aware Pervasive Systems: Architectures for a New Breed of Applications, CRC Press.
3. A. OSSEIRAN, J.F. MONSERRAT, P. MARSCH, (Eds.), 5G Mobile and Wireless Communications Technology, Cambridge University Press, 2016.

Reference Books

1. R. KAMAL (2008), Mobile and Pervasive Computing, Oxford University Press, 3rd Edition.
2. L. MERK, M. NICLOUS (2006), Principles of Mobile and Pervasive Computing, Dreamtech Press, 2nd Edition.
3. G. AGGELOU (2004), Mobile Ad hoc Networks: From Wireless LANs to 4G Networks, McGraw-Hill Professional.

Self-learning Material:

1. J.J. DRAKE, Z. LANIER, C. MULLINER, P.O. FORA, S.A. RIDLEY, G. WICHERSKI (2014), Android Hacker's Handbook, John Wiley, 1st Edition.
2. P. SINGH, NPTEL, IIIT Delhi: https://nptel.ac.in/courses/106106147/
3. Relevant research papers from MobiSys, Sensys, MC2R, MobiCom and UbiComp.

Course Title	**Edge and Fog Computing**	Course No.	**CS4xx**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech	Type	Elective
Prerequisite	Operating Systems, Computer Networks		

Objectives
This course will introduce design concepts, frameworks, and applications in Edge Computing to the audience

Learning Outcomes:
1. To understand various edge devices and their ecosystems, issues and challenges
2. To develop edge-based distributed computing platforms and applications

Contents
Introduction of Edge and Fog Computing: Internet of Things (IoT) and New computing paradigms, Fog computing: A platform for Internet of Things and analytics, Emergence of edge computing, Legal aspects of operating IoT applications in the fog. (6 Lectures)
Edge Architecture: Multi-Tier cloud computing framework; Data services with clouds at home; Leveraging mobile devices to provide cloud service at the edge; Fast, scalable and secure onloading of edge functions. (8 Lectures)
Networking for Edge & Fog: Integrating IoT + Fog + Cloud Infrastructures: System modeling and research Challenges, Management and Orchestration of network slices in 5G, Fog, Edge, and Clouds. (6 Lectures)
System Design: Optimization problems in fog and edge computing, Middleware for fog and edge Computing: Design issues, A Lightweight container middleware for edge cloud architectures. (8 Lectures)
Data Processing: Data management in fog computing, Predictive analysis to support fog application deployment, Using machine learning for protecting the security and privacy of Internet of Things (IoT) systems, fog Computing realization for Big data analytics. (8 Lectures)
Applications and Case Studies: Fog computing realization for Big data analytics, exploiting fog computing in health monitoring, Smart surveillance video stream processing at the edge for real-time human objects tracking, Fog computing model for evolving smart transportation applications. (6 Lectures)

Text Book
R. BUYYA, S.N. SRIRAMA (2019), Fog and Edge Computing: Principles and Paradigms, Wiley-Blackwell, 2019.

Course Title	**Virtualization and Cloud Computing**	Course No.	**CS7xx**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech	Type	Elective
Prerequisite	Operating Systems, Computer Networks		

Objectives

This course will introduce fundamentals of virtualization and concepts, frameworks, and applications in Cloud Computing to the audience

Learning Outcomes

1. To understand various virtualization techniques and their features and limitations
2. To understand design and concepts of a cloud computing framework

Contents

Cloud Computing: Concept, Definition, Cloud Types and Service Deployment Models. (5 Lectures)
Virtualization: Concept, Definition, Types of Virtualization, Hardware Virtualization, Full and Para Virtualization, Hypervisors, Hardware-assisted virtualization, operating system level virtualization, application virtualization (10 Lectures)
Virtual and Physical Networking: Introduction, vSwitches, virtual NICs, Virtual Networking, virtual LAN (5 Lectures)
Storage Virtualization: Introduction, SAN/NAS versus storage virtualization. (4 Lectures)
Virtual Machine Management: Base Virtual Machine, Virtual CPUs, Sockets, Cores, Memory Scaling Up and Scaling Down, USB Support, Virtual Disks, Live Migration. Security. (10 Lectures)
Containers: Concept, Definition, Docker, Container versus Virtualization, Portability, Remote deployment (5 Lectures)
Applications and Case Studies: Linux KVM, VirtualBox, Openstack. (3 Lectures)

Text Book

1. D.E. SARNA (2010), Implementing and Developing Cloud Computing Applications, CRC Press.
2. B.S. SODHI (2017), Topics in Virtualization and Cloud Computing, Ropar PB India, 2017.

Reference Book

B. FURHT, A. ESCALANTE (2010), Handbook of Cloud Computing (Vol. 3), Springer.

Course Title	**Compiler Design**	Course No.	**CSL7xx0**
Department	CSE	Structure (L-T-P-D[C])	3-0-0-0 [3]
Offered for	B.Tech., M.Tech. Ph.D.	Type	Elective
Prerequisite			

Objectives

1. To introduce key challenges for modern compilers and runtime systems
2. To explain various optimization techniques of a typical compilation workflow.

Learning Outcomes

1. To learn about compilation techniques for obtaining high performance on modern computer architectures.
2. To analyze and optimize various components of compilation stages

Contents

Representing Programs: Abstract Syntax Tree, Control Flow Graph, Dataflow Graph, Static Single Assignment, Control Dependence Graph, Program Dependence Graph, Call Graph. (11 Lectures)
Analysis/Transformation Algorithms: Dataflow Analysis, Interprocedural analysis, Pointer analysis, Rule-based analyses and transformations. Constraint-based analysis. (12 Lectures)
Applications: Scalar optimizations, Loop optimizations, Object-oriented optimizations, Register allocation, Program verification, Bug finding. (8 Lectures)
Advanced Topics: Just-in-time compilation, Memory Management, EDGE architecture compilation, Power-aware compilation, Machine specific optimizations (11 lectures).

Text Books

K.D. COOPER, L.TORCZON (2011), Engineering a Compiler, Morgan Kaufmann, 2nd Edition.
S. MUCHNICK (1998), Advanced Compiler Design Implementation, Morgan Kaufmann.

Reference Books

1. Y.N. SRIKANT, P.SHANKAR (Eds.) (2007), The Compiler Design Handbook: Optimizations and Machine Code Generation, CRC Press, 2nd Edition.
2. A.V. AHO, M.S. LAM, R. SETHI, J.D. ULLMAN (2007), Compilers: Principles, Techniques and Tools, Addison-Wesley, 2nd Edition.
3. R. ALLEN, K. KENNEDY (2001), Optimizing Compilers for Modern Architectures: A Dependence Based Approach, Morgan Kaufmann, 1st Edition.
4. M. WOLFE (1995), High-Performance Compilers for Parallel Computing, Addison-Wesley.
5. IEEE and ACM Digital Library.

Self Learning Material

NPTEL Lecture: https://nptel.ac.in/courses/106/108/106108052/

Course Title	**Software Maintenance**	Course No.	**CS4xx**
Department	Computer Science & Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech	Type	Elective
Prerequisite	Software Engineering		

Objectives

1. To explain theories, models, tools, and processes for the maintenance and evolution of large software systems and cloud applications
2. To familiarize with practices, techniques and processes that can help team build high quality software

Learning Outcomes

1. Understand the difficulties of developing code in a change context as opposed to new development
2. Identify the principal issues associated with software evolution and explain their impact on the software lifecycle
3. Describe techniques, coding idioms and other mechanisms for implementing designs that are more maintainable

Contents

Introduction: Basic concepts, difference between new development and maintenance, categorizing software change and limitations, maintenance frameworks (4 lectures)

Understanding Software & Evolution: Program comprehension models: top-down, bottom-up, opportunistic; Variants of software maintenance, relations to software engineering life-cycle, techniques of software evolution and testing, quantitative analyses, empirical analyses, qualitative analyses of software, evolution dynamics (10 Lectures)

Software Maintenance: Maintenance processes: quick-fix model, boehm's model, osborne's model, iterative enhancement model, process maturity, Search-based Software Testing, Summarization Techniques for Code, Changes, and Testing, Change impact analysis (7 Lectures)

Software Reuse: Reverse engineering levels & techniques: forward, restructuring, refactoring, reengineering, software reuse and evolution, the laws of software evolution, reuse techniques and design for reuse, libraries vs. application frameworks - Software product lines (7 Lectures)

Laboratory

Assignments/Projects on program comprehension and reverse engineering: program slicer, static analyser, dynamic analyzer, dataflow analyser, cross-reference, dependency analyser, transformation tools; Testing tools: simulators, test case generator, test path generators; source code, control system.

Text Books

1. Grubb, P., & Takang, A. A. (2003). Software maintenance: concepts and practice. World Scientific.
2. Galin, D. (2018). Software quality: concepts and practice. John Wiley & Sons..

Reference Book

Pigoski, T. M. (1996). Practical software maintenance: best practices for managing your software investment. Wiley Publishing.

Online Courses

Reverse Engineering Essentials: Udemy, https://www.udemy.com/course/reverse-engineering-essentials/

Network and Security

1. Introduction to wireless ad hoc network 7xx
2. 5G Mobile Networks 7xx
3. Cryptography and Network Security 4xx
4. Software Defined Networks (CS/EE) 7xx
5. Introduction to Blockchain 7xx
6. Intelligent Radio Networks (EE) 7xx
7. Data Communication Networks (EE) 3xx
8. Cognitive Internet of Vehicles (EE) 7xx
9. Delay Tolerant Networks (EE) 7xx

Course Title	**Introduction to Wireless Ad hoc Network**	Course No.	**CS7XX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3-0-0 [3]
Offered for	B.Tech, MTech, PhD	Type	Elective
Pre-requisites	Computer Networks		

Objectives

The course is intended to provide students with an understanding of the wide applicability of Wireless Ad-Hoc networks and their associated design issues and challenges.
It also sheds light on protocol design, mainly at the MAC layer.

Learning Outcomes

At the end of this course, students would be able to appreciate the mechanisms and interventions required at different levels to make Wireless Ad-hoc networks work in practice.
They would be able to explain different MAC protocols and their issues.
Students would also be able to contrast transport layer design issues arising out of mobility, with the conventional TCP.

Contents

Introduction to Wireless Ad-hoc Networks: Overview about Mobile ad-hoc networks (MANETs), Vehicular Ad-hoc networks (VANETs), Wireless Sensor Networks (WSNs) and Wireless Mesh Networks (WMNs). (2 Lectures)
Physical layer: Modulation techniques, Channel models, Case-study of 802.11a PHY. (3 Lectures)
Link Layer: Single-hop MAC protocols, multi-hop MAC protocols, Error correcting codes. (5 Lectures)
Network Layer: Mobile IP, Distributed wireless routing algorithms (AODV, DSDV, DSR, OLSR), Routing metrics. (8 Lectures)
Transport Layer: TCP over wireless, Transport level mobility management, Multihop transport protocols. (6 Lectures)
Introduction to 5G networks (3 Lectures)
Low Power Wide Area Networks (LPWAN) for IoT: LoraWAN, NB-IoT, SigFox. (9 Lectures)
Personal Area Networks: Bluetooth. (3 Lectures)
Future trends in wireless networks: 802.11ax, 802.11ay, mm-Wave WiFi. (3 Lectures)

Note

Please note that the course material would not be taken from a single book or resource. It would be a combination of textbook material, research papers and other sources.

Textbooks

W. STALLINGS (2009), Wireless Communications & Networks, Pearson Education India, 2nd Edition.
T.S. RAPPAPORT (2010), Wireless Communications: Principles and Practice, Prentice Hall, 2nd Edition.

Reference Books

C.S.R. MURTHY, B.S. MANOJ (2004), Ad hoc Wireless Networks: Architectures and Protocols, Pearson.
G. AGGELOU (2004), Mobile ad hoc networks: from wireless LANs to 4G networks, McGraw-Hill Professional.
S. BASAGNI, M. CONTI, S. GIORDANO, I. STOJMENOVIC, (Eds.), Mobile Ad hoc Networking, John Wiley, 2004.
A. GOLDSMITH (2005), Wireless Communications, Cambridge university press.
Recent relevant RFCs, Internet drafts, selected research papers from relevant venues: Mobicom, MobiSys, SIGCOMM, Infocom, IEEE TMC, ACM MC2R.

Self-learning Material

S. MISHRA NPTEL, IIT Kharagpur: https://nptel.ac.in/courses/106105160/
Relevant research papers.

Course Title	**5G Mobile Networks**	Course No.	**CS7XX**
Department	Computer Science and Engineering	Structure (L-T-P-C)	3-0-0
Offered for	B.Tech	Type	Core
Pre-requisites	Computer Networks, Ad-hoc Wireless Networks		

Objectives
The course is intended to provide students with an understanding of the forthcoming 5G technology, and its ability to transform the world for the better.

Learning Outcomes
Students would be able to appreciate the increasing role that SDNs play in 5G systems and beyond.

Contents
4G LTE networks: Introduction, overview and architecture. (4 Lectures)
From 4G to 5G : Why, When and How? (1 Lecture)
5G overview: Understanding 5GPP & NGMN, 5G architecture and design objective, 5G spectrum requirements, SDN in the 5G context. (6 Lectures)
5G Core: 5G RAN & Dynamic CRAN, Virtual Network Functions in the 5G core, 5G NR Logical Architecture, 5G Protocol Stack. (13 Lectures)
5G signalling: Millimeter wave propagation, Distributed massive MIMO principle, 5G Ultra dense cellular networks, 5G Coordinated Multi-Point. (8 Lectures)
5G Mobile Edge computing & Fog computing (2 Lectures)
5G applications: Healthcare, Transportation, Smart cities, AR, VR. (2 Lectures)
5G vs mm-wave Wifi: A comparison. (2 Lectures)
India's 5G policy and vision (1 Lecture)
The 6G vision (1 Lecture)
Paper discussions (2 Lectures)

Note
Please note that the course material would not be based on a single book or resource. It would be a combination of textbook material, research papers and other sources from the Internet.

Textbooks
A. OSSEIRAN, J.F. MONSERRAT, P. MARSCH (Eds.) (2016), 5G Mobile and Wireless Communications Technology, Cambridge University Press.

Reference Books
A. ELNASHAR, M.A. EL-SAIDNY, M. SHERIF (2014), Design, Deployment and Performance of 4G-LTE Networks: A Practical Approach, John Wiley & Sons.

Self-learning Material
Relevant research papers.

Title	**Cryptography and Network Security**	Course No.	**CS 4XXX**
Department	Computer Science and Engineering	Structure (L-T-P [C])	3–0–0 [3]
Offered for	B.Tech	Type	Elective
Prerequisite	Maths for Computing and Computer Networks		

Objectives

The course deals with the underlying principles of Network security. It develops the mathematical tools required to understand cryptography.
Provide fundamental understanding of the concepts related to information theoretic security, cryptographic techniques and protocols.

Learning Outcomes

The students will have the ability to:
Design encryption techniques for information and network security.
Evaluate the security of communication systems, networks and protocols based on a multitude of security metrics.

Contents

Introduction to Information Security: Schematic of secure communication systems, simple cryptosystems, Cryptanalysis – attack models, attacks on different ciphers. (3 Lectures)
Shannon's Approach to Cryptography: Perfect secrecy, entropy, One-time pad. (2 Lectures)
Symmetric Key Cryptography: Data Encryption Standard (DES) and differential cryptanalysis, The Advanced Encryption Standard (AES). (5 Lectures)
Cryptographic Hash Functions: Definition of hash functions and properties, Unkeyed hash functions, Keyed hash functions, Message Authentication Codes (MAC), The Random Oracle Model (ROM). (4 Lectures)
Authentication: A simple authentication protocol and possible attacks, Strong password protocols, BM Encrypted Key Exchange (EKE), Key Distribution Centers (KDC), Certification authorities and certificate revocation, KDC based authentication protocols. (6 Lectures)
Public Key Cryptosystems**:** RSA public key cryptosystem, The ElGamal public key cryptosystem and discrete logs. (4 Lectures)
Digital Signatures: The Digital Signature Algorithm (DSA), Key distribution and Key agreement protocols, Key pre-distribution, Diffie-Hellman key exchange. (8 Lectures)
Protocols: TCP/IP threats, The IPSEC protocol, The SSL and TLS protocols, Firewalls and Virtual Private Networks (VPNs), Electronic mail security, Worms, DDoS attacks, BGB and security considerations. (10 Lectures)

Text Book

W. STALLINGS (2018), Network Security Essentials: Applications and Standards, Pearson, 6th Edition.

Reference Books

D. BONEH, V. SHOUP (2017), A Graduate Course in Applied Cryptography (Version 0.4).
M.B. PATERSON, D.R. STINSON (2018), Cryptography: Theory and Practice, CRC Press, 4th Edition.
C. KAUFMAN, R. PERLMAN (2016), Network Security: Private Communication in a Public World, Pearson, 2nd Edition.

Self-Learning Material

Network security, NPTEL Course Material Department of Computer Science and Engineering, IIT Kharagpur: https://nptel.ac.in/courses/106105031/

Title	**Introduction to Blockchain**	Course No.	CSL 7XX
Departmen t	Computer Science and Engineering	Structure (L-T-P [C])	3–0–0 [3]
Offered for	B.Tech, MTech, PhD.	Type	Elective
Prerequisite	**Cryptography and**Network Security		

Objectives

The Instructor will:

Explain how blockchain technology works

Integrate blockchain technology into the current business processes to make them secure

Learning Outcomes

The students are expected to have the ability to:

Understand what and why of Blockchain

Explore major components of Blockchain and Identify a use case for a Blockchain application

Create their own Blockchain network application

Contents

Introduction to Blockchain: Digital Trust, Asset, Transactions, Distributed Ledger Technology, Types of network, Components of blockchain (cryptography, ledgers, consensus, smart contracts). (5 Lectures)

PKI and Cryptography: Private keys, Public keys, Hashing, Digital Signature. (6 Lectures)

Consensus: Byzantine Fault, Proof of Work, Proof of Stake. (6 Lectures)

Cryptocurrency: Bitcoin creation and economy, Limited Supply and Deflation, Hacks, Ethereum concept and Ethereum classic. (10 Lectures)

Hyperledger Fabric: Hyperledger Architecture, Membership, Blockchain, Transaction, Chaincode, Hyperledger Fabric, Features of Hyperledger, Fabric Demo. (8 Lectures)

Blockchain Applications: Building on the Blockchain, Ethereum Interaction - Smart Contract and Token (Fungible, non-fungible), Languages, Blockchain-as-a-service. (6 Lectures)

Textbook

A. BAHGA, V. MADISETTI (2017), Blockchain Applications: A Hands-On Approach, VPT.

Self Learning Material

M. SWAN (2015), Blockchain: Blueprint for a New Economy, O'Reilly Media.

R. WATTENHOFER (2016), The Science of the Blockchain, CreateSpace Independent Publishing Platform.

I. BASHIR (2017), Mastering blockchain, Packt Publishing Ltd.

K.E. LEVY, Book-smart, Not Street-smart: Blockchain-based Smart Contracts and the Social Workings of Law, Engaging Science, Technology, and Society, vol. 3, pp. 1-15, 2017.

Preparatory Course Material

MIT Online Blockchain Course, Learn Blockchain Technology: https://getsmarter.mit.edu/

Course Title	**Software Defined Networks**	Course No.	**CS7XXX**
Department	Computer Science and Engineering/Electrical Engineering	Structure (L-T-P-C)	2-0-0
Offered for	B.Tech	Type	Elective
Pre-requisites	Computer Networks		

Objectives
The course is intended to provide students with an understanding of the basics of SDNs.

Learning Outcomes
The course would enable students to look forward to SDN applications such as in datacenters and 5G systems.

Contents
Networking basics: Switching, Addressing, Routing (2 lecture)
Switching Architecture: Data, control and management planes, hardware lookup, forwarding rules, dynamic forwarding tables, autonomous switches and routers (4 lectures)
SDN Architectures: Plane Separation, Simple Device and Centralized Control, Network Automation and Virtualization, Openness SDN Controllers, SDN Applications, Northbound and Southbound APIs (5 lectures)
OpenFlow: Switch-Controller Interaction, Flow Table, Packet Matching, Actions and Packet Forwarding, Extensions and Limitations (3 lectures)
Network Function Virtualization: SDN vs. NFV, OPNFV, Inline Network Functions, Service Creation and Chaining, NFV Orchestration (5 lectures)
Emerging SDN Models: Protocol Models, Controller Models, Application Models, SDN in Datacenters: Multitenancy, Failure Recovery, SDN in Internet eXchange Points (IXPs) (4 lectures)
Data Center Networking in the context of SDN (5 lectures)

Note
Please note that the course material would not be based on a single book or resource. It would be a combination of textbook material, research papers and other sources from the Internet.

Textbooks
P. GORANSSON, C. BLACK, T. CULVER (2016), Software Defined Networks: A Comprehensive Approach. Morgan Kaufmann.
K. GRAY, T.D. NADEAU (2016), Network Function Virtualization, Morgan Kaufmann.

Self-learning Material
Relevant research papers.

Artificial Intelligence, Machine Learning

1. Artificial Intelligence (AIDE)
2. Deep Learning (AIDE)
3. Dependable Artificial Intelligence (AIDE)
4. Resource-Constrained AI (AIDE)
5. Advanced Machine Learning (AIDE)
6. Advanced Artificial Intelligence (AIDE)
7. Information Retrieval (AIDE)
8. Social Networks (AIDE)
9. Soft Computing Techniques (AIDE)
10. Autonomous Systems (IDRP)
11. Ethics, Policy, Law and Regulation in AI (AIDE)
12. Planning and Decision making for Robots (IDRP) (700)

Speech, Text and Vision

1. Digital Image Processing (AIDE)
2. Computer Graphics (AIDE)
3. Video Processing (AIDE)
4. Computer Vision (AIDE)
5. Introduction to AR and VR (AIDE)
6. Advanced Biometrics (AIDE)
7. Data Visualization (AIDE)
8. Natural Language Understanding (AIDE)
9. Speech Understanding (EE)
10. Principles of Biological Vision and Applications 7xx

Title	**Principles of Biological Vision and Applications**	Number	CS7XXX
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech., M.Tech., Ph.D.	Type	Elective
Prerequisite	None		

Objectives
The instructor will provide:
An account of various biological perception and cognitive processes.
Computational models for such processes
Examples of machine vision applications with such computational models.

Learning Outcomes
On completion of the course, the students are expected to have the ability to:
Appreciate the principles of biological perception and cognition.
Apply computational models for perception and cognition processes of biological vision in various machine vision tasks.

Contents
(*Fractal 1* (14 Lectures))
Introduction: Perception, reasoning and cognition; roles of knowledge, memory and learning. (2 Lectures)
Eye and early vision: Lateral Inhibition; convolution; detection of oriented edges; color, texture and motion perceptions; peripheral vision. (4 Lectures)
Reasoning systems: Deductive, abductive and inductive reasoning; statistical property of nature; Bayesian framework of reasoning; Bayesian networks; parameter estimation; on complexity of models and prior probabilities; information integration; hierarchical Bayesian models and inductive generalization. (8 Lectures)
(*Fractal 2* (14 lectures))
Late Vision: Depth perception; perceptual grouping; foreground-background separation; multi-stability; models for object recognition; hierarchical models; visual quality and aesthetics. (5 Lectures)
Visual attention: cognitive, information-theoretic, Bayesian, context based and object-based attention models; evaluation of attention models. (6 Lectures)
Introduction to cognitive architectures: Active vision; perceive-interpret-act cycles; modeling cognition processes; roles of short-term and long-term memory; review of STAR architecture. (3 Lectures)
(*Fractal 3* (14 lectures))
Knowledge Representation and learning: Role of knowledge in visual interpretation; memory, knowledge and learning; short-term and long-term memory; structured and unstructured knowledge; semantic networks; frame-based representation; symbol grounding problem; perceptual knowledge; unified framework for conceptual and perceptual knowledge; learning in knowledge-based systems. (5 Lectures)
Neural networks for vision: Introduction to neural network; CNN for image classification and object detection; knowledge representation in neural networks; RNN and life-long learning; LSTM; recurrent attention models; reinforcement learning, meta-learning and multi-task learning; knowledge-infused learning; graph networks. (6 Lectures)
Applications in machine vision tasks: Computational photography; digital heritage; social robots; smart content delivery and re-purposing. (3 Lectures)

Textbook
H. GHOSH (In Press – expected Jun/Jul 2020*)*, Computational Models for Cognitive Vision,, Wiley-IEEE Press.
Reference Material
Research literature to be announced in class.

Social and Sustainable Computing

1. Social Computing (3-0-0)
2. Sustainable Computing (3-0-0)
3. ICT for Development (2-0-0)
4. Machine Learning in Epidemiology (3-0-0)
5. Health Informatics (3-0-0)
6. Computational Social Choice Theory (3-0-0)
7. Computational Cognition and Behavior Modelling (3-0-0)
8. Crowd-sourcing and human-computing (3-0-0)
9. Environmental Informatics (3-0-0)
10. Computational Microeconomics (3-0-0)

Title	**Social Computing**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	Data Structure and Algorithms		

Objectives

The Instructor will:

1. Introduce the fundamental and current challenges in social computing.
2. Explain how to design social computing systems to be effective and responsible.

Learning Outcomes

The students will have the ability to:

1. Explore social computing systems, get experience with social data analyses and focus on design, and evaluation of a social software as their final project for the course.
2. Engage in the creation of new computationally-mediated social environments.

Contents

Introduction and overview Social Software: *Introduction,* Different types of social software like blogging, microblogging, Q&A, Forum, Wiki etc. *[3 Lectures]*

Social information processing *:* Tagging, Social Navigation, Social Search, Social Bots *[4 Lectures]*

Group collaboration*:* Wikis and Wikipedia, Computer supported collaboration tools, Content sharing, Open source software development *[6 Lectures]*

Recommender systems *:* Content based, collaborative filtering, challenges of social information processing *[5 Lectures]*

Social data analysis *:* Visualization, Sense-making, APIs, Facebook, Wikipedia, Twitter *[6 Lectures]*

Social capital *:* Definitions and measures, Social capital and social networks, Role of online communities on social capital *[6 Lectures]*

Challenges of online communities*:* Dealing with newcomers, under-contribution problem Encouraging contributions to online communities, Strategies supported by social science theories *[8 Lectures]*

Social computing, cities, and society*: Impact on physical and psychological well-being, Connection to real world, Equality of participation [4 Lectures]*

Text Book

Becker, Howard S. Writing for social scientists: How to start and finish your thesis, book, or article. University of Chicago Press, 2008.

Reference Books

1. Bail et al (2018). Exposure to opposing views on social media can increase political polarization. Proceedings of the National Academy of Sciences, 115(37), 9216-9221.
2. Dubois, E., & Blank, G. (2018). The echo chamber is overstated: the moderating effect of political interest and diverse media. Information, Communication & Society, 21(5), 729-745.
3. Salganik, M. J., & Watts, D. J. (2009). Web-based experiments for the study of collective social dynamics in cultural markets. Topics in Cognitive Science, 1(3), 439-468.

Self-Learning Material

Social Computing, IIT Kharagpur, http://cse.iitkgp.ac.in/~pawang/courses/SC16.html

Title	**Sustainable Computing**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	PRML / Introduction to Machine Learning		

Objectives

The Instructor will:

Describe ways to leverage networking, sensing, and computational strategies to help address sustainability issues related to biodiversity, climate, environment, urban design, transportation, buildings and others. Draw from professional experiences to ground all topics and discussions in "real world" examples.

Learning Outcomes

The students will have the ability to:
Gain an understanding of how data and computation can be used to identify root causes of sustainability challenges, create targeted strategies to address them, engage stakeholders, and track and communicate progress.
Develop skills to design metrics and solutions for evaluation of progress towards sustainability goals.
Comprehend the power and shortcomings of data in communicating complex sustainability issues.

Contents

Introduction, SDG goals, & Sustainability metrics*:* Humanity and the environment, Sustainability, Dimensions of sustainable design, Sustainability metrics. *[3 Lectures]*

Climate: Computational techniques for weather predictions & models of climate variance. *[5 Lectures]*

Biosphere*:* Species distribution modelling; Aggregate phenomena modelling; Ecosystem modelling; Animal identification; Case study: Green Security games, Bird Returns program. [*6 Lectures*]

Health, Food & Water*:* Health/medical informatics & Telemedicine; Poverty mapping, Food & Farm optimization; Modelling for water distribution. [*6 Lectures*]

Insuring impact with technology for Education & Gender equality. *[3 Lectures]*

Usability & HCI for assistive technologies. *[3 Lectures]*

Green technology & E-waste. *[2 Lectures]*

Sustainable energy systems*:* Energy-constrained scheduling, Electricity demand & renewable resource prediction, Models for energy consumption, Smart grid. *[4 Lectures]*

IT energy efficiency*:* Measure energy efficiency of computer system components in sync with solution frameworks; Energy consumption estimation of whole systems; Energy efficiency metrics; Sustainable resource management techniques for cloud computing, edge/fog computing, high-performance computing, wearable computing, Internet-of-Things (IoT), and cyber-physical systems; Sustainable data centers. *[10 Lectures]*

Text Books

Theis, T., & Tomkin, J. (2015). Sustainability: A Comprehensive Foundation. OpenStax CNX.
Issa, T., & Isaias, P. (2015). Sustainable Design: HCI, Usability and Environmental Concerns. Springer-Verlag London.
Green & Sustainable Computing - Part I & II. Elsevier.

Reference Material

1. Gomes, C. et al. (2019). Computational Sustainability: Computing for a Better World and a Sustainable Future. Communications of the ACM, 62(9),56-65.
2. Sustainability Metrics Reading List: https://afddf8e8-2dfc-4526-9ba6-71e1899413f3.filesusr.com/ugd/571f98_b203aa22d378452b986dbc667acb13b7.pdf.
3. Reading List: http://www.cs.cornell.edu/courses/cs6702/2011sp/.
4. Reading List: https://www.coursera.org/learn/sustainability#syllabus

Title	**ICT for Development**	Number	CS L4XX0
Department	Computer Science and Engineering	L-T-P [C]	2-0-0 [2]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	None		

Objectives

The Instructor will:

1. Facilitate an interactive platform for students to discuss and learn ICT strategies for various problems.
2. Reflect upon his/her own experiences in deploying ICT solutions in social settings and the associated challenges.

Learning Outcomes

The students will have the ability to:

1. Appreciate the potential that ICT possesses to make an impact on the society.
2. Learn through various case studies, that different problem settings require different solution approaches.

Contents

From the origins of ICTD to Big data and development [3 lectures]
Development theory and the critiques of ICTD [4 lectures]
Localization techniques and Identity management [3 lectures]
ICTD: knowledge economies and development [3 lectures]
e-learning and development [2 lectures]
e-governance and development [2 lectures]
ICTs and Environment sustainability [2 lectures]
Building sustainable rural information service networks [2 lectures]
Sustainability failures of rural telecentres [1 lecture]
ICT for Agriculture [2 lecture]
ICTD: Information sharing practices [1 lecture]
ICTD for community networks and disaster management [2 lectures]
ICT4D: the future [1 lecture]

Text books

1. Heeks, R. (2017). *Information and communication technology for development (ICT4D)*. Routledge.
2. **References**
3. https://www.oii.ox.ac.uk/study/course/ICT_Development_Reading_List.pdf
4. http://act4d.iitd.ernet.in/act4d/index.php?option=com_content&view=article&id=33&Itemid=42
5. https://www.kth.se/student/kurser/kurs/kursplan/IV1009-20082.pdf?lang=en
6. https://www.manchester.ac.uk/study/masters/courses/list/06237/msc-icts-for-development/course-details/MGDI60701#course-unit-details
7. **Self-Learning Material**

Title	**Machine Learning in Epidemiology**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	Introduction to Machine Learning/ PRML		

Objectives
Course focuses on advances in machine learning and its application to causal inference and prediction via *Targeted Learning*, which allows the use of machine learning algorithms for prediction and estimating so-called causal parameters, such as average treatment effects, optimal treatment regimes, etc.

Learning Outcomes
On completion of the course the students will have:
1. Ability to apply machine learning algorithms to prediction problems and estimate and derive inference for the resulting fit.
2. Ability to use the fits of machine learning algorithms to estimate causal effects using simple substitution estimators.
3. Knowledge of How the general methodology applies to goals of Precision Medicine.

Contents

Introduction: Introduction to epidemiology, statistical methodologies in epidemiologic studies, curse of dimensionality in epidemiology, Relation between ML and epidemiology, more on epidemiology. *[4 Lectures]*

Prediction of Observational Epidemiology: Clinical Decision Support, Disease Surveillance, Introduction to Targeted Machine Learning. *[4 Lectures]*

Causal Inference: Understanding bias, Correlation vs. causalty, Causal graph inference. *[4 Lectures]*

Super-Learning: Parametric Models, Data Adaptivity, Performance Measure on Independent Data, Customizable Optimality criterion, Sequential super learning. *[8 Lectures]*

Target Update of Machine Learning: TMLE, HAL, HAL-TMLE, C-TMLE, TMLE with Clustering, On-line super learning, On-line targeted learning for time series. *[14 Lectures]*

Case Studies in Epidemiology: Ebola, Corona etc. *[4 Lectures]*

Future Directions: Deep learning in clinical epidemiology, Recurrent unit neural networks to identify individuals at risk. *[4 Lectures]*

Reference Books

1. Van der Laan & Rose (2018), Targeted Learning in Data Science Causal Inference for Complex Longitudinal Studies, Springer
2. Van der Laan & Rose (2011), Targeted Learning in Causal Inference for Observational and Experimental Data, Springer

Reference Articles

1. Timothy L. Wiemken, Robert R. Kelley (2020), Machine Learning in Epidemiology and Health Outcomes Research, *Annual Review of Public Health*, 41(1), 21-36
2. Goldstein, N. D., LeVasseur, M., & McClure, L. A. (2020). On the Convergence of Epidemiology, Biostatistics, and Data Science. *Harvard Data Science Review.*
3. Char DS, Shah NH, Magnus D. (2018), Implementing machine learning in health care- Addressing ethical challenges, *New England Journal of Medicine*; 378-981 (3).
4. Chiavegatto Filho ADP, Dos Santos HG, do Nascimento CF, Massa K, Kawachi I. (2018), Overachieving municipalities in public health: A machine learning approach, Epidemiology; 29-836-40.
   https://medium.com/causal-data-science/causal-data-science-721ed63a4027
5. Other relevant articles

Title	**Health Informatics**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	Data Structure and Algorithms		

Objectives

1. Develop knowledge about problems and challenges that health informatics addresses
2. To develop an understanding of different types of data exchange formats and standards techniques for different purposes.
3. To equip the students with various possible applications of the digital data analysis in healthcare.

Learning Outcomes

The students are expected to have the ability to:

1. Apply basic knowledge to the research and practice of health informatics
2. Demonstrate basic skills and knowledge in health informatics for application in future health-related careers
3. Utilize the tools and techniques for collecting, storing, securing, retrieving, and reporting health care data and information

Contents

Introduction: Overview, health data, information, and knowledge, electronic health records, system architecture for EHRs, ambulatory EHR functions and interoperability, personal health records and decision aids. *[6 lectures]*

Health Care Information Systems: Healthcare data standards - Overview, methods, protocols, terminologies, and specifications for the collection, exchange, storage, and retrieval of information associated with health care applications. *[6 lectures]*

Data Collection and Quality Assurance: Data collection, cleaning data, managing change, Using Data for Care Delivery, Coordination, and Quality Improvement, Quality assurance, Security, Privacy. *[8 lectures]*

Clinical Data: Introduction, medical digital data formats and exchange, medical image formats such as DICOM in the context of health informatics. *[4 lectures]*

Informatics: Information retrieval, bioinformatics - usage of informatics in genomics and other aspects of molecular biology, patient case history analytics, applications in public health, ethical issue in health informatics. *[8 lectures]*

Medical Diagnostic Decision Support: probabilistic approaches, clinical scores, logical approaches - expert system, inference engines, case studies. *[6 Lectures]*

Digital healthcare: Telemedicine, mobile for health, public health *[4 lectures]*

Text Books

1. Hoyt, R. E., & Yoshihashi, A. K. (2014). Health informatics: practical guide for healthcare and information technology professionals. Lulu. com
2. Hersh, W. R., & Hoyt, R. E. (2018). Health Informatics: Practical Guide Seventh Edition. Lulu. com

Reference Material

1. Landais, P., Boudemaghe, T., Suehs, C., Dedet, G., Lebihan-Benjamin, C., Venot, A., ... & Quantin, C. (2014). Medical Informatics, e-Health: Fundamentals and Applications.
2. Healthcare data standards, https://www.ncbi.nlm.nih.gov/books/NBK216088/
3. Data Collection and Quality Assurance, https://www.ncbi.nlm.nih.gov/books/NBK208601/
4. Computer-aided diagnosis medical image analysis techniques, https://www.intechopen.com/books/breast-imaging/computer-aided-diagnosis-medical-image-analysis-techniques

Self-Learning Material

Health Informatics Specialization, Coursera Online, https://www.coursera.org/specializations/health-informatics

Title	**Computational Social Choice Theory**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	Design and Analysis of Algorithms		

Objectives

The Instructor will:

1. Familiarize with mechanisms for collective decision making
2. Focus on interface of social choice theory with theoretical computer science
3. Enable students to conduct research in this fast-growing field

Learning Outcomes

The students will have the ability to:

1. Understand the problems and research directions in computational social choice.
2. Mathematically formulate the problems that arise in society, and design efficient rules for these problems.
3. Analyse computational complexity of problems in social choice.

Contents

Introduction to Voting Theory: social choice functions, voting rules and their axiomatic study. *[5 Lectures]*

***Computational hardness and Algorithms for voting rules*:** Condorcet, minimax, Dodgson's rule, and Kemeny. *[5 lectures]*

Manipulation in Voting: various means of manipulation: strategic voting, bribery, control, computational hardness and algorithms. *[7 lectures]*

Multiwinner Election: voting rules and their computational complexity. *[4 lectures]*

Matching under preferences: introduction to stable matching, Gale-Shapley algorithm, hospital/resident problem, stable roommate problem. *[8 lectures]*

Manipulation in Matching: means of manipulations, computational hardness and algorithms. *[3 lectures]*

Introduction to Fair Allocation: basic terminologies, fairness and efficiency criteria. *[4 lectures]*

Fair Allocation of goods: fair allocation of indivisible and divisible goods. *[6 lectures]*

Text Book

1. Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A.D. (2016). Handbook of Computational Social Choice. Cambridge University Press.
2. Gusfield, D., & Irving, R.W. (1989). The Stable Marriage Problem: Structure and Algorithms. MIT press
3. Manlove, D. (2013). Algorithmics Of Matching Under Preferences. World Scientific
4. **Reference Books**
5. Endriss, U. (2017). Trends in Computational Social Choice. Lulu. com
6. Lecture Notes on Fair Division by U. Endriss

Title	**Computational Cognition & Behavior Modelling**	Number	CS L4XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite	Probability & Statistics, PRML		

Objectives

The Instructor will:

1. Draw on formal models from classic and contemporary artificial intelligence to explore fundamental issues in human cognition and behavior.
2. Encourage students to explore answers to questions on cognition and behavior (e.g., What are the different forms that our knowledge of the world takes? How does this lead to behaviour?) and ways to model them
3. Help students develop an understanding of how computational modeling can advance cognitive science, how cognitive science can contribute to research in machine learning and AI, and how to fit and evaluate cognitive models to understand behavioral data.

Learning Outcomes

The students will have the ability to:

1. Demonstrate knowledge of basic concepts and methodologies of cognitive and behavioral modelling - be able to design simple models for sample problems.
2. Demonstrate understanding of the relationship between computational models and psychological theories - critically assess the psychological adequacy of a given model.
3. Qualitatively and quantitatively evaluate computational models of cognition and behavior.
4. Extend the understanding of these models to the design of architectures for social cognitive agents for applications such as autonomous driving, and assistive robotics.

Contents

Fractal 1 [Computational Modelling Techniques for Cognition and Behavior]

Introduction [1 Lecture]: The motivation underlying the computational modelling of cognition, and the possible questions that can be answered

Probabilistic approaches to modelling cognition & behavior [4 lectures]: Bayesian inference, hierarchical Bayesian models, probabilistic graphical models

Inductive Logic Programming & Language of thought models. *[3 Lectures]*

Deep learning, reinforcement learning for modelling cognition. *[3 Lectures]*

Information sampling & Active learning. *[3 Lectures]*

Fractal 2 [Cognition Modelling]

The mind as an information-processing system - Marr's levels of analysis (computational, algorithmic, implementation) [*1 Lecture*]

Structure and formation of intuitive theories of cognition of physical, biological and social systems; Learning causal relations. *[3 Lectures]*

Symbolic (rule-based), subsymbolic (probabilistic) & connectionist (network-based) models of cognition [*3 Lectures*]

Methodology and issues in the development and evaluation of cognitive models [7 Lectures]: Which psychological data are relevant? How do we collect such data (e.g., EEG, Eye-tracker; across ages & genders)? What predictions can be made by the models of cognition? How can these predictions be tested? - Some possible example models: language processing, memory, reasoning, categorization.

Fractal 3 [Behavioral Modelling]

Introduction [*4 Lectures*]: Brain & behavior; Understanding social behavior; General motivation underlying computational modelling of behavior

Symbolic (rule-based), subsymbolic (probabilistic) & connectionist (network-based) approaches to behavior modelling [*3 Lectures*]

Methodology and issues in the development and evaluation of behavioral models [7 Lectures]: Which behavioral data are relevant? How do we collect such data (e.g., EEG, Eye-tracker; across ages, genders & societies)? What predictions can be made? How do these predictions apply to assistive system design?

Text Books

1. Russell, S. J., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach. 4^{th} Edition. Prentice Hall.
2. Farrell, S., & Lewandowsky, S. (2018). Computational Modeling of Cognition and Behavior. Cambridge University Press.

Reference Material

1. Issa, T., & Isaias, P. (2015). Sustainable Design: HCI, Usability and Environmental Concerns. Springer-Verlag London.
2. McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. Nature Reviews Neuroscience, 4(4), 310-322.
3. Peterson, J., Abbott, J., & Griffiths, T. (2016). Adapting deep network features to capture psychological representations. Presented at the 38th Annual Conference of the Cognitive Science Society.
4. Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. Science, 331(6022), 1279-1285.
5. Griffiths, T.L., Lieder, F., & Goodman, N.D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. Topics in Cognitive Science, 7(2), 217-229.
6. Wilson, R.C. and Collins, A.G.E. (2019). Ten simple rules for the computational modeling of behavioral data. eLife 2019, 8:e49547.
7. Goodman, N. D., Tenenbaum, J. B., & Gerstenberg, T. (2014). Concepts in a probabilistic language of thought. Center for Brains, Minds and Machines (CBMM).

Title	**Crowd-sourcing and human-computing**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Program Elective
Prerequisite	Social Computing, Data Structures & Algorithms		

Objectives

The Instructor will:

1. familiarize with the techniques to manage the crowd
2. focus on building intelligent systems that involve a combination of computers and humans collaborating seamlessly over the Internet

Learning Outcomes

The students will have the ability to:

1. Program the crowd
2. Apply usability principles for designing crowd tasks that elicit high-quality responses
3. Use statistical methods to improve the quality of the work received
4. Build systems that interface with crowd labor in real time

Contents

Introduction to Crowdsourcing and Human Computation: introduction, history, connections to social computing, collective intelligence, and management science [4 lectures]
Human Computation Algorithms: design patterns, case study, discussion [4 lectures]
Usability Engineering for CrowdSourcing: input-agreement mechanisms, output-agreement mechanisms [4 lectures]
Gold Standards: consensus algorithms, incentive mechanisms, gamification [6 lectures]
Real-time crowdsourcing: queuing theory for predicting worker availability, ensemble learning, crowd agents [6 lectures]
Crowdsourcing Subjective Tasks: social Q&A, expertise in crowdsourcing, local crowdsourcing / community sourcing, crowd funding, citizen science [6 lectures]
Task Routing: push and pull approaches [4 lectures]
Crowds & Applications: Crowds: mechanical turk, social media; Applications: computer vision, NLP [8 lectures]

Text Book

Law, E., & Ahn, Ahn, L. (2012). Human Computation: An Integrated Approach to Learning from the Crowd, Morgan & Claypool Publishers.

Reference Books

1. Ghosh, A., Lease, M., (2016). Human Computation and Crowdsourcing. Proceedings of the Fourth AAAI Conference
2. Lease M., Alonso O. (2018) Crowdsourcing and Human Computation: Introduction. In: Alhajj R., Rokne J. (eds) Encyclopedia of Social Network Analysis and Mining. Springer, New York.

Self-Learning Material

Title	**Environmental Informatics**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All)	Type	Elective
Prerequisite			

Objectives

The Instructor will:

1. Introduce techniques for environmental modeling, environmental databases and information systems.

Learning Outcomes

The students will have the ability to:

1. Apply the techniques for environment modeling on typical applications of environment monitoring and protection.
2. Appreciate the multidisciplinary nature of environmental problems.

Contents

Introduction to Environmental Informatics: environmental objects, characterisation of an environmental system, environmental metadata. [2 lectures]

Data capture and data storage: introduction, object taxonomies, mapping the environment, advanced techniques. [3 lectures]

Environmental sampling: sampling design, essentials for sampling of environmental data, satellite imagery [4 lectures]

Hydrological Modeling: Rainfall-Runoff models, Precipitation Models, Climate Models. [4 lectures]

Geographical information systems [4 lectures]

Environmental data analysis: Scales of operation of environmental data, re-sampling of environmental data, approximation of environmental data, trend estimation of environmental data [7 lectures]

Environmental statistics: probability distributions of environmental data, statistical measures, statistical tests, regression and correlation [6 lectures]

Environmental time series: fourier analysis, stationary processes, time correlation functions, frequency functions [7 lectures]

Environmental simulation models: modelling procedure, types and classification of mathematical models, process identification, an eutrophication simulator. [5 lectures]

Text Book

1. Günther, O., (1989). Environmental Information Systems. Springer, Berlin.
2. Box, G. E. P., Jenkins, G. M. & Reinsel, G. C., (1994). Time Series Analysis. 5th ed., Prentice Hall, Englewood Cliffs.

Reference Books

Avouris, N. M. and Page, B. (eds.), (1995). Environmental Informatics – Methodology and Applications of Environmental Information Processing. Kluwer, Dordrecht.

Title	**Computational Microeconomics**	Number	CS L7XX0
Department	Computer Science and Engineering	L-T-P [C]	3–0–0 [3]
Offered for	B.Tech (All Branches)	Type	Program Elective
Prerequisite	Design and Analysis of Algorithms		

Objectives

The Instructor will:

1. Familiarize with the market design
2. Focus on algorithms to clear the market
3. Familiarize with the algorithmic game theory

Learning Outcomes

The students will have the ability to:

1. Design market
2. Use computer science knowledge to solve economic problems

Contents

Introduction to Market Design: introduction to various types of auctions, bidding languages, matching market, voting. [6 lectures]

Auction: Myerson's Lemma, Knapsack auction, revenue-maximising auction, exchanges. [8 lectures]

Matching Market: Kidney exchange, stable matching, various notions of optimal matching. [8 lectures]

Noncooperative Game Theory: self-interested agents, price of anarchy, non-atomic and atomic models of selfish routing. [5 lectures]

Games in Normal Form: pareto optimality, nash equilibria, various solution concepts and their computation. [10 lectures]

Games in Extensive Form: backwards induction, subgame perfect equilibrium, imperfect information, equilibrium refinements, computing equilibria. [5 lectures]

Text Book

Shoham, Y. & Brown, K.L., (2008), Multiagent systems: Algorithmic, game-theoretic, and logical foundations, Cambridge University Press.

Reference Books

Roughgarden, T. (2016), Twenty lectures on algorithmic game theory, Cambridge University Press.

Self Learning Material

https://www.youtube.com/watch?v=TM_QFmQU_VA