Computer Science Engineering

S.No	New Course code	Name of Course	L-T-P-C	Proposed Level (UG/PG)
1	CS 601	Software Development for Scientific Computing	3-0-0-6	PG
2	CS 603	Approximation algorithms	3-0-0-6	PG
3	CS 604	Parametrized Algorithms and Complexity	3-0-0-6	PG
4	CS 605	Reinforcement Learning	3-0-0-6	PG
5	CS 606	Advanced Topics in Embedded Computing	3-0-0-6	PG
6	CS 607	Advanced Computer Networks	3-0-0-6	PG
7	CS 608	FPGA for communication networks prototyping	3-0-0-6	PG
8	CS 609	Software Defined Networking (SDN) and Network Function Virtualization (NFV)	3-0-0-6	PG
9	CS 610	Advanced Distributed Systems	3-0-0-6	PG
10	CS 611	Advanced Software Systems Lab	0-1-6-8	PG
11	CS 612	Statistical Pattern Recognition Laboratory	0-0-3-3	PG
12	CS 614	Reinforcement Learning Laboratory	0-0-3-3	PG
13	CS 616	Statistical Pattern Recognition	3-0-0-6	PG
14	CS 617	Special Topics in Hardware Systems	3-0-0-6	PG
15	CS 620	Formal Models for Concurrent and Asynchronous Systems	3-0-0-6	PG
16	CS 621	Logic and Applications	3-0-0-6	PG
17	CS 622	Special Topics in Automata and Logics	3-0-0-6	PG
18	CS 623	Advanced Topics in Communication Networks	3-0-0-6	PG
19	CS 624	Compilers - Principles and Implementation	3-0-0-6	PG
20	CS 625	Topics in Stochastic Control and Reinforcement Learning	3-0-2-8	PG
21	CS 626	Topics in Data Structures and Algorithms	2-0-2-6	PG
22	CS 627	Data Structures	3-0-0-6	PG
23	CS 628	Algorithms	3-0-0-3	PG
24	CS 629	Introduction to Reinforcement Learning	2-0-2-6	PG
25	CS 630	Statistical Machine Learning	2-0-2-6	PG
26	CS 631	Seminar	0-0-4-4	PG
27	CS 632	Runtime Verification	3-0-0-6	PG

Computer Science Engineering

28	CS 702	Systems Bootcamp for ML	1-0-2-4	PG
29	CS 703	Topics in Data Structures and Algorithms	3-0-0-6	PG
30	CS 704	Advanced Algorithms	3-0-0-6	PG
31	CS 705	Topics in Graph Theory	3-0-0-6	PG
32	CS 706	Topics in Parameterized Algorithms and Complexity	3-0-0-6	PG
33	CS 801	Power Aware Computing	3-0-2-8	PG
34	CS 802	Dataflow Processor Architecture (Guided Study)	3-0-0-6	PG
35	CS 810	Advanced Computer Architecture	3-0-3-9	PG
36	CS 438	Natural Language Processing	3-0-0-6	PG
37	CS 439	Introduction to Sanskrit Computational Linguistics	3-0-0-6	PG
38	CS 440	Advances in Cloud Technologies	1-0-0-2	PG
39	CS 636	Advanced Data Structures and Algorithms	0-0-3-3	PG
40	CS 633	Advanced Data Structures and Algorithms	3-0-0-6	PG
41	CS 618	Advanced Software Development Laboratory	1-0-4-6	PG
42	CS 638	Combinatorics and Probability	3-0-0-6	PG
43	CS 639	Scalable Data Mining	3-0-0-6	PG

44	CS 101	Computer Programming	3-0-2-8	UG
45	CS 103	Introduction to Programming – 1	3-0-2-4	UG
46	CS 104	Introduction to High Performance Computing	0.5-0-0-1	UG
47	CS 106	Data Structures and Algorithms	3-0-0-6	UG
48	CS 111	Data Structures and Algorithms Laboratory	0-0-3-3	UG
49	CS 202	Automata Theory	3-1-0-8	UG
50	CS 203	Discrete Structures	3-0-0-6	UG
51	CS 205	Design and Analysis of Algorithms	3-0-0-6	UG
52	CS 209	Artificial Intelligence	3-0-0-6	UG
53	CS 213	Software Systems Laboratory	1-3-0-8	UG
54	CS 214	Artificial Intelligence Lab	0-0-3-3	UG
55	CS 301	Computer Architecture	3-0-0-6	UG
56	CS 303	Data Bases and Information Systems	3-0-0-6	UG
57	CS 304	Operating Systems	3-0-0-6	UG

Computer Science Engineering

58	CS 306	Introduction to Artificial Neural Networks	3-0-0-6	UG
59	CS 309	Research and Development Project	6 credits	UG
60	CS 311	Computer Architecture Laboratory	0-0-3-3	UG
61	CS 313	Data Bases and Information Systems Laboratory	0-0-3-3	UG
62	CS 314	Operating Systems Laboratory	0-0-3-3	UG
63	CS 315	Computer Networks Laboratory	0-0-3-3	UG
64	CS 316	Compilers Lab	0-0-3-3	UG
65	CS 320	Research and Development Project II	6 credits	UG
66	CS 323	Compilers	3-0-0-6	UG
67	CS 324	Programming Techniques	3 credits	UG
68	CS 348	Computer Networks	3-0-0-6	UG
69	CS 401	Software Engineering	0-0-3-3	UG
70	CS 402	Distributed Systems	3-0-0-6	UG
71	CS 403	Graph Theory and Combinatorics	3-0-0-6	UG
72	CS 405	B.Tech Project-CSE	6 credits	UG
73	CS 409	B.Tech Project-CSE - II	6 credits	UG
74	CS 410	Parallel Computing	3-0-0-6	UG
75	CS 421	Logic for Computer Science	3-0-0-6	UG
76	CS 422	Principles of Programming Languages	3-0-0-6	UG
77	CS 424	Elements of Programming	2-0-2-6	UG
78	CS 426	Introduction to Blockchains	3-0-0-6	UG
79	CS 427	Mathematics for Data Science	3-0-0-6	UG
80	CS 428	Deep Learning	3-0-0-6	UG
81	CS 430	Computer Graphics	3-0-2-8	UG
82	CS 432	Introduction to Logic	3-0-0-6	UG
83	CS 433	Cloud Software Development	3-0-0-3	UG
84	CS 434	Programming Parallel Machines	3-0-0-6	UG
85	CS 435	Introduction to Abstractions and Paradigms for Programming	1-0-0-2	UG
86	CS 436	Operational Analysis	3-0-0-6	UG
87	CS 437	Guided Study	6 credits	UG
88	CS440	Advanced in Cloud Technologies	1-0-0-2	UG
89	CS 707	Advanced Topics in Deep learning	3-0-2-8	PG

Computer Science Engineering

90	CS 708	Adversarial Machine Learning	3-0-0-6	PG
91	CS 709	Machine Learning Applications in Wireless Networks	2-1-0-6	PG

Computer Science Engineering

1	Title of the course (L-T-P-C)	Software Development for Scientific Computing (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to Data Structures and Algorithms, C / C++ / Java / Matlab
3	Course content	Algorithmic Patterns in Scientific Computing: dense and sparse linear algebra, structured and unstructured grid methods, particle methods (N-body, Particle-Particle, Particle-in-cell, Particle-in-a- mesh), Fast Fourier Transforms, Implementing PDEs, C++ standard template library (STL), Introduction to debugging using GDB, GMake, Doxygen, Version Control System, Profiling and Optimization, asymptotic analysis and algorithmic complexity. Mixed-language programming using C, Fortran, Matlab, and Python, Performance analysis and high-performance code, Data locality and auto tuning, Introduction to the parallel programming world.
4	Texts/References	- Stroustrup C++ Language Reference (https://www.stroustrup.com/4th.html) - Suely Oliveira, David Steward: Writing Scientific Software: A Guide to Good Style. Cambridge University Press, 2006 - Web references to GNU Make, GDB, Git, GProf, Gcov. - Code Complete: A Practical Handbook of Software Construction - https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS -2006-183.html

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Approximation algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Data Structures and Algorithms (CS201)
3	**Course content**	Introduction, approximation schemes, design and analysis of approximation algorithms - combinatorial algorithms, linear programming based algorithms. Hardness of approximation.
4	**Texts/References**	**Textbook**: Approximation algorithms. Vazirani, Vijay V. Berlin: springer, 2001. **Reference:** The design of approximation algorithms. Williamson, David P., andDavid B. Shmoys. Cambridge university press, 2011.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Parametrized Algorithms and Complexity** **(3-0-0-6)**
2	Pre-requisite courses(s)	Data Structures and Algorithms, Design and Analysis of Algorithms
3	Course content	Introduction. Kernelization, Bounded Search Trees, Iterative Compression, Treewidth, Advanced kernelization algorithms. Lower bounds: Fixed- parameter intractability, lower bounds based on ETH, lower bounds for kernelization. **Parameterized Algorithms, Kernelization, and Complexity of Graph Modification Problems**
4	Texts/References	**Textbook:** • Parameterized Algorithms, Marek Cygan, Fedo • V. Fomin, Lukasz Kowalik. Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Sourabh. Springer. 2015 **Reference:** • Parameterized Complexity, R. G. Downey, and M. R. Fellows. Springer Science and Business Media. 2012

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Reinforcement Learning** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Basic Probability and Linear Algebra
3	**Course content**	Bandit Algorithms -- Regret based - UCB, Thomson Sampling, PAC Based - Median Elimination, Markov Decision Process Modeling - Bellman Equation, Dynamic Programming Solutions - Value and Policy Iteration, Linear Programming, Model free methods - Monte Carlo and Temporal Difference Methods - Q-learning, Value function Approximation - State Aggregation, Critic Only/Value Based Methods Methods - TD methods, Q- Learning, SARSA, Actor Only/Policy Based methods - Reinforce, Actor-Critic Methods - Policy Gradient, Natural Actor Critic, Deep RL - DQN, A3C, Model Based RL, Integrating Learning and Planning, Case-studies.
4	**Texts/References**	1. Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3. Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4. Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Reinforcement Learning 2** (3-0-0-6)
2	**Pre-requisite courses(s)**	CS 301
3	**Course content**	Introduction to systems software in embedded platforms Boot loader Embedded Linux kernel (Processes, Threads, Interrupts) Device Drivers Scheduling Policies (including Real Time) Memory Management Optimizations (Data level and Memory level) Embedded Systems Security Introduction to Embedded GPUs and Accelerators Embedded Heterogenous Programming with Open CLApplication Case Study on Embedded Platforms – eg. Neural Network inferencing on Embedded Platforms, Advanced Driver Assistance Systems
4	**Texts/References**	1. Building Embedded Linux Systems, 2nd Edition by Gilad Ben-Yossef, Jon Masters, Karim Yaghmour, Philippe Gerum, O'Reilly Media, Inc. 2008 2. Linux Device Drivers, Third Edition By Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, O'Reilly Media, Inc. 2005 3. Embedded Systems: ARM Programming and Optimization by Jason D Bakos, Elsevier, 2015 4. Learning Computer Architecture with Raspberry Pi by Eben Upton, Jeff Duntemann, Ralph Roberts, Tim Mamtora, Ben Everard, Wiley Publications, 2016 5. Real Time Systems by Jane S. Liu, 1 edition, Prentice Hall; 2000 6. Practical Embedded Security: Building Secure Resource-Constrained Systems by Timothy Stapko, Elsevier, 2011

Computer Science Engineering

1	Title of the course (L-T-P-C)	Advanced Computer Networks (3-0-0-6)
2	Pre-requisite courses(s)	
3	Course content	1. Circuit, Packet and Virtual Circuit Switching, MPLS 2. Switch Architectures, Buffering Strategies, Input and Output Queuing, IP Buffer Sizing 3. Quality of Service and Scheduling Algorithms 4. IP Address Lookup and IP Packet Classification algorithms 5. Software Defined Networking 6. Next Generation Network Architectures, Network Provisioning and Design, and "Green" (Energy- Efficient) Networking 7. Data Driven Networking 8. Wireless Networks - MANETs, Sensor Networks, Cellular Networks, Personal Area Networks 9. Content Based Delivery Networks - Principles of data dissemination, aggregation and caching that are applied to sensor networks, Internet of Things, and other content-based paradigms. Students will survey recent research publications on opportunistic networks and next generation content-based networking ideas. 10. Delay tolerant Networks 11. Network security - authentication, access control, privacy preservation, intrusion detection and prevention 12. Performance analysis of new Networking ideas using simulation (such as Network Simulator (ns3), GENI testbed, Simulink, Open LTE and Open C-RAN frameworks)
4	Texts/References	**Textbook:** Computer Networks: A Systems Approach, Larry Peterson and Bruce Davie, 2011. Performance Evaluation of Computer Systems, by Raj Jain, Wiley, 1991. Computer Networking, Kurose and Ross, Addison-Wesley, 2012. **Reference:** 1. An Engineering Approach to Computer Networking by S. Keshav, 1997, Addison-Wesley Professional Series. 2. Network Routing, by Deepankar Medhi and Karthikeyan Ramasamy, Morgan Kaufmann, 2007. 3. SDN: Software Defined Networks, by Thomas D. Nadeau, Ken Gray, O'Reilly Media, 2013. 4. High Performance Switches and Routers, By H.Jonathan Chao and Bin Liu, Wiley, 2007. Network Algorithmics, by George Varghese, Morgan Kaufmann, 2005

Computer Science Engineering

1	Title of the course (L-T-P-C)	**FPGA for communication networks prototyping** **(3-0-0-6)**
2	Pre-requisite courses(s)	EE 224 Digital System Exposure on Computer Network
3	Course content	History and evaluation of FPGAs; FPGA architecture; Introduction to Quartus Prime (vendors and design tools; vendors and programmable logic); Exploiting Simulation tools (e.g., ModelSim); Exploiting FPGAs for multi-domain technologies; Introduction to radio access networks-fronthaul (e.g., common public radio interface); optical network; metro and core networks; Cross-layer design; The role of FPGA in the specified network segments and use case scenarios; In and Out; Clocks and Registers; State Machines; Modular Design; Memories Managing Clocks; I/O Flavors; Exploiting Qsys and Nios II tools.
4	Texts/References	1. C. Maxfield, "The Design Warrior's Guide to FPGAs: Devices, Tools and Flows", Jun. 2004, eISBN 9780080477138 2. FPGAs For Dummies, 2nd Intel Special Edition. Published by. John Wiley & Sons, Inc 3. William J. Dally, R. Curtis Harting, "Digital Design: A Systems Approach 1st Edition", Cambridge University Press, September 2012, ISBN 9780521199506 4. Verilog by Example: A Concise Introduction for FPGA Design, Blaine C. Readler 5. Course materials: Slides; Notes; Tutorials from Altera website https://www.altera.com/support/training/university/materials-tutorials.html 6. R. Ramaswami, K. Sivarajan, G. Sasaki; "Optical Networks: A Practical Perspective," 3rd Ed., Morgan Kaufmann, ISBN: 9780123740922.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Software Defined Networking (SDN) and Network Function Virtualization (NFV) (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to Computer Networks
3	Course content	History and evolution of SDN; SDN Architecture (Application, Control, Infrastructure Layer); SDN Interfaces (East/West/North/South-bound interfaces); SDN Security; SDN routing; SDN standards; SDN Controllers; Network Operating Systems and Languages; OpenFlow; Software Switches (e.g. OpenVSwitch); SDN Simulation/Emulation Platforms (e.g. Mininet); Federated SDN networks; SDN Applications and Use Cases; Programming assignment/project; Need for NFV; NFV and SDN Relationship; Virtual Network Functions; Service Function Chaining; NFV Specifications; NFV Architecture; NFV Use Cases; NFV Management and orchestration (MANO); Open-source NFV; Hands-on exercises based on OpenStack/Docker.
4	Texts/References	● Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black, Morgan Kaufmann Publications, 2014 ● SDN – Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013 ● Software Defined Networking with OpenFlow, By Siamak Azodolmolky, Packt Publishing, 2013 ● Gray, Ken, and Thomas D. Nadeau. Network function virtualization. Morgan Kaufmann, 2016. ● Zhang Ying. Network Function Virtualization: Concepts and Applicability in 5G Networks. John Wiley & Sons, 2018. ● Foundations of modern networking- SDN, NFV, QoE, IoT, and Cloud, William Stallings ● James Kurose and Keith Ross, "Computer Networking, A Top-Down Approach"

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Advanced Distributed Systems** **(3-0-0-6)**
2	Pre-requisite courses(s)	Operating Systems, Data Structures and Algorithms, Programming in C++
3	**Course content**	Synchronization, Global Snapshot and Distributed Mutual Exclusion, Consensus & Agreement, Checkpointing & Rollback Recovery, Deadlock Detection, Termination Detection, Message Ordering & Group Communication, Fault Tolerance and Self-Stabilization, Peer to Peer Systems Mining Data Streams in a distributed system: filtering data streams, queries on streams, pattern detection Key-Value Storage: Cassandra, HBase Virtualization and Cloud Computing: virtual machines containers Message oriented communication, Publish Subscribe Systems (use case Apache Kafka) Security: Distribution of security mechanisms, access control, and security management.
4	**Texts/References**	Distributed Computing: Principles, Algorithms, and Systems- Ajay D. Kshemkalyani and Mukesh Singhal Mining Massive data sets- Jure Leskovec, Anand Rajaraman, Jeff Ullman Distributed Algorithms – An Intuitive Approach (The MIT Press) by Wan Fokkink Distributed Algorithms-Nancy Lynch

Computer Science Engineering

1	Title of the course (L-T-P-C)	Advanced Software Systems Lab (0-1-6-8)
2	Pre-requisite courses(s)	Nil
3	Course content	The course should involve 3 large programming projects. A set of three project oriented assignments which will be announced at the start of each semester with definite submission deadlines. The set of assignments will be designed to develop skills and familiarity with a majority of the following: make, configuration management tools, installation of software, archiving and creation of libraries, version control systems, documentation and literate programming systems, GUI creation, distributed state maintenance over a network, programming in different environments like desktop and handhelds, program parsing and compilation including usage of standard libraries like pthreads, numerical packages, XML and semi-structured data, simulation environments, testing and validation tools.
4	Texts/References	• Unix concepts and applications by Sumitabha Das • PThreads Programming A POSIX Standard for Better Multiprocessing By Dick Buttlar, Jacqueline Farrell, Bradford Nichols • Head-First Python by Paul Barry

Computer Science Engineering

1	Title of the course (L-T-P-C)	Statistical Pattern Recognition Laboratory (0-0-3-3)
2	Pre-requisite courses(s)	Currently taking statistical pattern recognition theory course
3	Course content	The lab will closely follow the theory course. The idea is to have the students implement the basic algorithms on different topics studied in the statistical pattern recognition theory course.
4	Texts/References	1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. 2. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Reinforcement Learning Laboratory (0-0-3-3)
2	Pre-requisite courses(s)	Currently taking reinforcement learning theory course
3	Course content	The lab will closely follow the theory course. The idea is to have the students implement the basic algorithms on different topics studied in the reinforcement learning theory course.
4	Texts/References	1. Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3. Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4. Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5. Regret Analysis of Stochastic and Nonstochastic Multi- armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Statistical Pattern Recognition (3-0-0-6)
2	Pre-requisite courses(s)	Multivariate Calculus and Linear Algebra, Probability, Programming
3	Course content	Bayesian Decision Making and Bayes Classifier, Parametric and Non Parametric Estimation of Densities, General Linear Models, Discriminative Learning based Models, Dimensionality Reduction Techniques, Empirical and Structural risk minimization, Ensemble Methods - Bagging, Boosting, Pattern Clustering, Graphical Models, Statistical Learning Theory.
4	Texts/References	- R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. - C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Special Topics in Hardware Systems** **(3-0-0-6)**
2	Pre-requisite courses(s)	Digital Systems Computer Architecture
3	Course content	Introduction Processors Memories Special Processors and Accelerations Architecture for Machine Learning Detailed Syllabus is attached in the appendix
4	Texts/References	1. J .L. Hennessy, D. A. Patterson: Computer Architecture :A Quantitative Approach(The Morgan Kaufmann Series in Computer Architecture and Design),2011 2. J. P. Shen, M. H. Lipasti: Modern Processor Design: Fundamentals of Superscalar Processors , Waveland Press, 2013 3. B. Reagan, R. Adolf, P. Whatmough, G.Y-Wei, D.Brooks:Deep Learning for Computer Architects Synthessis Lectures on Computer Architecture, Morgan & Claypool,2017.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Logic and Applications** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Mathematics, Theory of computation
3	**Course content**	**Module 1:**Propositional Logic:Natural deduction, semantics, soundness, completeness, compactness, normal forms, Horn clauses and satisfiability. **Module 2:** Predicate Logic: Natural deduction, resolution, undecidability, expressiveness. **Module 3:** Some decidable fragments of first-order logic and their decision procedures: propositional logic, equality with uninterpreted functions, linear arithmetic, Presburger logic ,bit vectors, arrays, pointer logic. **Module 4:** SAT and SMT solvers: theory and practice: Decision procedures for combinations of first-order theories: Nelson-Oppen, Shostak, Satisfiability Modulo Theories(SMT)Combination with SAT solvers: eager, lazy approaches. Student is required to do a small project using a SAT/SMT solver.
4	**Texts/References**	- Logic in Computer Science, Michael Huth and Mark Ryan, Cambridge University Press. - Mathematical Logic for Computer science, Mordechai Ben-Ari, Springer. - Logic for Computer Scientists, Uwe Schoning, Birkhauser. - SAT/SMT by example, Dennis Yurichev.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Special Topics in Automata and Logics (3-0-0-6)
2	Pre-requisite courses(s)	Discrete Mathematics, Theory of computation, Logic and its applications.
3	Course content	This course aims at giving an introduction to the theory of automata working on infinite words and infinite trees and connections thereof to logics. These automata and related logics are of fundamental importance in the areas of formal specification and verification of reactive systems. If time permits, we will also discuss some basic results in finite model theory. Below is a list of topics, which will be discussed in this course. Automata on finite words - equivalence of MSO and automata; Automata on infinite words â€" different acceptance conditions; Closure properties and equivalence of different acceptance conditions and related translations Determinization and complementation results; Equivalence of automata and MSO and decidability of MSO; Automata on infinite trees - different acceptance conditions; Closure properties and comparison of expressive power of different acceptance conditions and related translations Complementation result for tree automata via parity games; Equivalence of MSO and tree automata; Decidability of MSO over tress; Parity games and determinacy; Ehrenfeucht-Frasse games in logics and applications.
4	Texts/References	1. Wolfgang Thomas: Automata on infinite objects, Handbook of theoretical computer science (vol B): formal methods and semantics, Elsevier. 2. Wolfgang Thomas: Languages, automata, and logic, Handbook of formal languages, vol. 3: beyond words, Springer-Verlag. 3. Dominique Perrin, Jean-Eric Pin: Infinite words, Elsevier 4. Erich Gradel, Wolfgang Thomas, Thomas Wilke: Automata, logics, and infinite games: a guide to current research. LNCS, Springer-Verlag. 5. Leonid Libkin: Elements of finite model theory, Springer-Verlag.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Advanced Topics in Communication Networks** **(3-0-0-6)**
2	Pre-requisite courses(s)	Undergraduate Computer Networks course, Good Programming Background.
3	Course content	1. 4G/5G Networks – Radio Access Architecture, Evolved Packet Core, Protocols, Network Management Algorithms, Network Optimization, Resource Allocation Algorithms, Security. 2. Fog Computing, Edge Computing – Architecture, Optimization, Resource Allocation, and Load Balancing 3. Internet of Things 4. Data Driven Networking 5. Application of SDN and NFV in next generation IoT/cellular networks.
4	Texts/References	Research papers and online courses from Coursera/Udacity/Edx will be referred to for learning the afore-mentioned topics.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Compilers - Principles and Implementation (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to Data Structures and Algorithms, Computer Architecture, Automata Theory
3	Course content	Structure of a compiler, the compiled and interpreted execution models. Lexical analysis and parsing using lex and yacc. Scope and visibility analysis. Data layout and lifetime management of data. Runtime environment. Dynamic memory allocation and Garbage collection. Translation of expressions, control structures, and functions. Code generation and local optimizations, Lattice theory, register allocation, instruction scheduling, optimizations - dataflow, control flow, reaching definitions, and liveness analysis, code transformation-tiling, fusion.
4	Texts/References	1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D.Ullman: Compilers: Principles, Techniques, and Tools, 2/E, AddisonWesley 2007. 2. Andrew Appel: Modern Compiler Implementation in C/ML/Java, Cambridge University Press, 2004 3. Dick Grune, Henri E. Bal, Cerial J.H. Jacobs and Koen G. Langendoen: Modern Compiler Design, John Wiley & Sons, Inc. 2000. 4. Michael L. Scott: Programming Language Pragmatics, Morgan Kaufman Publishers, 2006. 5. Fisher and LeBlanc: Crafting a Compiler in C.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Topics in Stochastic Control and Reinforcement Learning (3-0-2-8)
2	Pre-requisite courses(s)	Probability, Linear Algebra and Multi-variable Calculus
3	Course content	Bandit Algorithms -- Regret based - UCB, Thomson Sampling, PAC Based - Median Elimination, Optimality of Bandit Algorithms, Finite Horizon Problems, Infinite Horizon Problems - Total Cost Criterion, Discounted Cost Criterion, Average Cost Criterion, Markov Decision Process (MDP), Partially observable MDP (POMDP), and Dynamic Programming Solutions - Value and Policy Iteration, Model free methods - Monte Carlo and Temporal Difference Methods - Q-learning, SARSA, on/off-policy learning, Stochastic Approximation: Single and multi-timescale stochastic approximation, ordinary differential equation based convergence results. Convergence of SARSA and Q- learning, Value function Approximation - State Aggregation, Critic Only/Value Based Methods - TD methods, gradient temporal difference learning, Actor Only/Policy Based methods - Reinforce, Actor-Critic Methods - Policy Gradient, Natural Actor Critic, Deep RL - DQN, A3C, Regret Based RL - Upper Confidence Reinforcement Learning, Posterior sampling Reinforcement Learning.
4	Texts/References	1. Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 4th edition, 2012. 2. Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 3. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 4. Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 5. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Topics in Data Structures and Algorithms (2-0-2-6)
2	Pre-requisite courses(s)	No prerequisites
3	Course content	Data structures - arrays, linked lists, stacks and queues, heap and binary search tree. Algorithm design techniques - divide and conquer, greedy, and dynamic programming. Algorithms for graph problems. Asymptotic notations, Complexity lower bounds and NP-completeness.
4	Texts/References	**Textbook:** 1. Cormen, Leiserson, Rivest and Stein, **Introduction to Algorithms**, 3rd edition, by, MIT Press, 2009. **Reference:** 1. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, McGraw Hill Education, 2008. 2. Kleniberg and Tardos, **Algorithm Design**, 1st edition, Pearson, 2006.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Data Structures** **(3-0-0-3)**
2	**Pre-requisite courses(s)**	None
3	**Course content**	● Asymptotic notations-Theta, Big-oh, Big-omega, little-oh, little-omega ● Data structures - arrays, linked lists, stacks and queues, heap and binary search tree, hash tables

4	**Texts/References**	**Textbook**:
		  1. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. MIT press; 2009.
		Reference:
		  1. Brass P. Advanced data structures. Cambridge: Cambridge University Press; 2008.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Algorithms** **(3-0-0-3)**
2	**Pre-requisite courses(s)**	
3	**Course content**	Algorithm design techniques - divide and conquer, greedy, and dynamic programming, Algorithms for graph problems. Complexity, lower bounds, and NP-completeness.
4	**Texts/References**	**Textbook** 1. Cormen TH, Leiserson CE, Rivest RL,Stein C. Introduction to algorithms. MIT press; 2009. **Reference**: 1. Dasgupta S, Papadimitriou CH, Vazirani UV. Algorithms. New York: McGraw-Hill Higher Education; 2008..

Computer Science Engineering

1	Title of the course (L-T-P-C)	Introduction to Reinforcement Learning (2-0-2-6)
2	Pre-requisite courses(s)	Exposure to Probability Models, Stochastic Process and its Applications, Math for Data Science I and II
3	Course content	andit Algorithms -- Regret based - UCB, Thomson ampling, Markov Decision Process Modeling - Bellman quation, Dynamic Programming Solutions - Value and olicy Iteration, Model free methods - Monte Carlo and emporal Difference Methods - Q-learning, Value unction Approximation - State Aggregation, Critic nly/Value Based Methods Methods - TD methods, -Learning, SARSA, Actor Only/Policy Based methods - einforce, Actor-Critic Methods - Policy Gradient, Deep L - DQN, A3C.
4	Texts/References	1. Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. 2. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. 3. Bertsekas, Dimitri P. Dynamic Programming and Optimal Control. Vol. 1 and 2. 4th edition, 2012. 4. Algorithms for Reinforcement Learning, Csaba Szepesvári, Morgan & Claypool, 2009. 5. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems, Sébastien Bubeck and Nicolò Cesa-Bianchi, Foundations and Trends in Machine Learning, Volume 5, Number 1, 2012.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Statistical Machine Learning (2-0-2-6)
2	Pre-requisite courses(s)	Exposure to Probability Models, Stochastic Process and its Applications, Math for Data Science I and II
3	Course content	Bayesian Decision Making and Bayes Classifier, Parametric Estimation of Densities, General Linear Models, EM Algorithm, Discriminative Learning based Models, Dimensionality Reduction Techniques, Empirical risk minimization, Ensemble Methods - Bagging, Boosting, Clustering
4	Texts/References	1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001. 2. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Runtime Verification** **(3-0-0-6)**
2	Pre-requisite courses(s)	Not applicable
3	Course content	**1**. Overview of Runtime Verification, and its comparison with other Formal Verification approaches. **2**. Fundamentals: Propositional and First-Order Logic, Temporal Logics (Linear and Metric) **3**. Propositional LTL and its variants: specification of properties, runtime verification strategies, expressibility, and monitorability. **4**. First Order LTL and its variants: specification of properties, runtime verification strategies, expressibility, and monitorability. **5**. Discussion of various state-of-the-art tools and case studies.
4	Texts/References	**1.** K. Havelund, D. Peled, "Runtime Verification: From Propositional to First Order Temporal Logic", Tutorial at International Conference on Runtime Verification, 2018. **2.** Ezio Bartocci, Yliès Falcone. "Lectures on Runtime Verification". Springer, 2018. ISBN: 978-3-319-75632-5 **3.** Michael Huth, Mark Ryan, "Logic in Computer Science: Modelling and Reasoning about Systems", Cambridge University Press, 2004. ISBN: 978-0521543101 **4.** Research publications on Runtime Verification.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Systems Bootcamp for ML** **(1-0-2-4)**
2	**Pre-requisite courses(s)**	Exposure to basics of computer programming.
3	**Course content**	Systems Programming lab focuses on programming principles and skills for building systems software. It also focuses on how to build and optimize Al/ML models from a systems perspective. ● Introduction to UNIX, shell programming, version control and management (SVN/Git) ● Introduction to Libraries (Boost, STL) ● Introduction to Profiling (Performance analysis at system level and application level), ML perf ● Introduction to various DL frameworks and DL Inference engines, Hardware backends ● Database fundamentals and programming ● GPU Programming (CUDA, OpenCL)/ ● Automatic Code Generation - TVM Stack ● Mobile-edge cloud computing (Computational offloading decisions) ● Programming assignments/projects will be given related to the above topics.
4	**Texts/References**	1. Unix Man Pages for all unix tools, **Advanced Bash Scripting Guide from the Linux Documentation Project** (www.tldp.org) 2. Loeliger, Jon and MCullough, Matthew**, Version Control with Git: Powerful tools and techniques for collaborative software development**, O'Reilly, 2012. 3. Sommerville, Ian. **Engineering Software Products: An Introduction to Modern Software Engineering**, Pearson, May 2019. 4. Rodriguez Andres : **Deep Learning Systems: Algorithms, Compilers, and Processors** for Large- scale Production, Morgan and Claypool publishers, 2020

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Topics in Design and Analysis of Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Mathematics, Design and Analysis of algorithms, Data structures and Algorithms.
3	**Course content**	**Module 1:** <u>Iterated Improvement Paradigms</u>- Computational and Algorithmic Thinking, Matching Algorithms, Flow Algorithms (16 hours). **Module 2:** <u>Approximation Algorithms</u>- Greedy Approximation, Local Search, Linear Programming, Duality Techniques (16 hours) **Module 3:** <u>Randomized Algorithms-</u> Monte Carlo and Las Ve-gas types, Randomized Attrition, Randomized In- cremental Design, Sampling, Chernoff type bounds and High Confidence Analysis, Abundance of witness for Monte Carlo algorithms, Number theoretic Algorithms (16 hours).
4	**Texts/References**	1. [**OA**] James B. Orlin, Ravindra K. Ahuja, and Thomas L. Magnanti, "Network Flows", Prentice Hall, 1993. 2. [**WS**] David P. Williamson and David B. Shmoys, "The Design of Approximation Algorithms", CambridgeUniversity Press, 2011. 3. [**MR**] Rajeev Motwani and Prabhakar Raghavan, "Ran-domized Algorithms", Cambridge University Press, 1995.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Advanced Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Discrete Mathematics, Design and Analysis of algorithms, Data structures and Algorithms
3	**Course content**	**Module 1:** <u>Iterated Improvement Paradigms</u>- Computational and Algorithmic Thinking, Matching Algorithms, Flow Algorithms (16 hours). **Module 2:** <u>Approximation Algorithms</u>- Greedy Approximation, Local Search, Linear Programming, Duality Techniques (16 hours) **Module 3:** <u>Randomized Algorithms</u>- Monte Carlo and Las Ve-gas types, Randomized Attrition, Randomized In- cremental Design, Sampling, Chernoff type bounds and High Confidence Analysis, Abundance of witness for Monte Carlo algorithms, Number theoretic Algorithms (16 hours).
4	**Texts/References**	**1**. [**OA**] James B. Orlin, Ravindra K. Ahuja, and Thomas L. Magnanti, "Network Flows", Prentice Hall, 1993. **2**. [**WS**] David P. Williamson and David B. Shmoys, "The Design of Approximation Algorithms", CambridgeUniversity Press, 2011. **3**. [**MR**] Rajeev Motwani and Prabhakar Raghavan, "Ran-domized Algorithms", Cambridge University Press, 1995.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Topics in Graph Theory** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Discrete Structures
3	**Course content**	Recap: fundamental concepts. Topics in factors, covering and packing, cuts, connectivity, coloring, planarity, perfect graphs, Ramsey theory, and random graphs.
4	**Texts/References**	**Textbook:** 1. Introduction to Graph Theory (2nd Edition), Douglas B. West. Prentice Hall. **References:** 1. Algorithmic Graph Theory and Perfect Graphs (2nd Edition), Martin Charles Golumbic. Elsevier. 2. Graph Theory (Graduate Texts in Mathematics, 5th Edition), Reinhard Diestel. Springer.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Topics in Parameterized Algorithms and Complexity (3-0-0-6)
2	Pre-requisite courses(s)	Data Structures and Algorithms, Design and Analysis of Algorithms
3	Course content	Introduction. Kernelization, Bounded Search Trees, Iterative Compression, Treewidth, Advanced kernelization algorithms. Lower bounds: Fixed-parameter intractability, lower bounds based on ETH, lower bounds for kernelization. Parameterized Algorithms, Kernelization, and Complexity of Graph Modification Problems.
4	Texts/References	**Textbook:** 1. Parameterized Algorithms, Marek Cygan, Fedor V. Fomin, Lukasz Kowalik. Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Sourabh. Springer. 2015. **Reference:** 1. Parameterized Complexity, R. G. Downey, and M. R. Fellows. Springer Science and Business Media. 2012.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Power Aware Computing** **(3-0-2-8)**
2	**Pre-requisite courses(s)**	Exposure to Computer Architecture, Operating Systems
3	**Course content**	Introduction to Power and Energy, Power consumption modeling and estimation, Dynamic power management and DVFS, Leakage reduction techniques, circuit-level and Micro-architecture techniques, Power states and ACPI support, Memory/cache power optimizations. Software level techniques, GPU power modeling and optimizations.
4	**Texts/References**	1. S. Kaxiras, M. Martonosi, Computer Architecture Techniques for Power-Efficiency, Synthesis Lectures on Computer Architecture. Morgan &C laypool publishers. 2. Siva G. Narendra, Anantha Chandrakasan 3. P. Leakage in Nanometer CMOS Technologies, Series on Integrated Circuits and Systems 4. Rakesh Chadha, J Bhasker an ASIC Low Power Primer: Analysis, Techniques and Specification.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Dataflow Processor Architecture (Guided Study) (3-0-0-6)
2	Pre-requisite courses(s)	Computer Architecture
3	Course content	The philosophy of dataflow; static and dynamic approaches; contrast with conventional out-of-order pipelines; understanding different granularities of operations, appreciating the performance and power possibilities; understanding the caveats; studying particular example architectures; analyzing the fundamental concept in the light of modem trends in the semiconductor industry; analyzing the fundamental concept in the context of particular application classes.
4	Texts/References	Papers(list not exhaustive); 1 Exploring the potential of heterogeneous von Neumann/Dataflow execution models, Nowatzkietal., ISCA2015. 2 Dataflow Machine Architecture. AH Veen,ACM Computing Surveys.1986. 3 Dataflow Architecture: Are dataflow computers commercially viable?, Kavi et al.,IEEE Pontentials 1992. 4 Synchronous Dataflow Architectures for Network processors, Carlstrom et al., IEEE Micro,2004. 5 Dataflow architectures,Culler.Annual review of Computer Science 1986. 6 An architectural comparison of dataflow systems .Srini. dataflow Computing: Theory and Practice 1992. 7 The Machester Prototype Dataflow Computer.Gurd et al.ACM.1985. 8 Monsoon: an Explicit Token-store Architecture.Papadopolous et al. ACM SIGARCH 1990.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Advanced Data Structures and Algorithms Lab** (0-0-3-3)
2	**Pre-requisite courses(s)**	None
3	**Course content**	Concepts on advanced data structures and algorithms used in practical life. The lab course is to implement all these data structures and algorithms using any programming language, preferably one among C/C++, Python, and Java.
4	**Texts/References**	1. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to algorithms, 2022. MIT press. 2. Online lecture notes by Jeff Erickson [https://jeffe.cs.illinois.edu/teaching/algorithms/] 3. 3. Steven Skiena, The Algorithm Design Manual, 2000, Springer. 4. 4. J. Kleinberg and E. Tardos, Algorithm Design, 2005, Pearson 5. 5. Mark Weiss, Data structures and algorithm analysis in C++ (Java), 2014, Pearson

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Advanced Data Structures and Algorithms** (3-0-0-6)
2	**Pre-requisite courses(s)**	None
3	**Course content**	**Module 1:** Basics: asymptotic notations, recurrences, basic data structures **Module 2**: Advanced data structures: heaps, priority queues, hash tables, data structures based on trees **Module 3:** Design paradigms and complexity analysis: divide and conquer, dynamic programming, greedy algorithms, amortized analysis **Module 4**: Advanced topics: graph algorithms, string algorithms, geometric algorithms, complexity lower bounds, coping up with hard problems.
4	**Texts/References**	1. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to algorithms, 2022. MIT press. 2. Online lecture notes by Jeff Erickson [https://jeffe.cs.illinois.edu/teaching/algorithms/] 3. 3. Steven Skiena, The Algorithm Design Manual, 2000, Springer. 4. 4. J. Kleinberg and E. Tardos, Algorithm Design, 2005, Pearson 5. 5. Mark Weiss, Data structures and algorithm analysis in C++ (Java), 2014, Pearson.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Advanced Software Development Laboratory** (1-0-4-6)
2	**Pre-requisite courses(s)**	None
3	**Course content**	Editing: Vim/emacs, Presentation: latex, beamer, Build, Integration and Deployment, Version Control, and Documentation: make, GitHub and git, doxygen Programming: HTML, CSS, Shell scripting, AWK, SED Exploring features of IDE (e.g. eclipse, vscode) using a high-level programming language such as Java, Python, C++, Debugging (using gdb, using IDE). Unix/Linux basics: shell, file system, permissions, process hierarchy, process monitoring, ssh, scp, rsync, grep, find, head, tail, tar, cut, sort, I/O redirection, pipes Profiling tools: (e.g., gprof, prof, perf, valgrind)
4	**Texts/References**	1. Article title: The Linux Documentation Project, URL: www.tldp.org, Website title: The Linux Documentation Project, Date accessed: March 22, 2024. 2. Article title: Python Tutorial, URL: http://docs.python.org/3/tutorial/index.html, Website title: Python Documentation, Date accessed: March 22, 2024. 3. Article title: Oracle Java Documentation, URL: http://docs.oracle.com/javase/tutorial/, Website title: The Java Tutorials, Date accessed: March 22, 2024. 4. Latex - A document preparation system, 2/e, by Leslie Lamport, Addison-Wesley, 1994. 5. Article title: GNU Make, URL: https://www.gnu.org/software/make/manual/html_node/index.html#Top, Website title: GNU Make, Date accessed: March 22, 2024. 6. Article title: Book, URL:Git - Book (git-scm.com), Website title: Git, Date accessed: March 22, 2024. 7. Article title: GDB: The GNU Project Debugger, URL: GDB Documentation (sourceware.org), Website title: GDB Documentation, Date accessed: March 22, 2024. 8. Article title: Valgrind User Manual, URL:Valgrind, Website title: Valgrind, Date accessed: March 22, 2024. 9. Article title: Overview, URL: https://www.doxygen.nl/manual/index.html, Website title: Doxygen, Date accessed: March 22, 2024. 10. URL: GNU gprof, Website title: GNU gprof, Date accessed: March 22, 2024.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Combinatorics and Probability (3-0-0-6)
2	Pre-requisite courses(s)	None
3	Course content	Combinatorics: **Principles of counting** : Rule of sum, rule of product, permutations, combinations. Partition, modular arithmetic. Double counting, generating functions. Binomial coefficients, binomial theorem, multinomial theorem **Probability theory:** Probability axioms and laws, random variables, binomial distribution, Poisson, exponential and normal distributions, expectations and moments, joint distribution, conditional distribution , conditional expectations, convergence of random variables, law of large numbers, central limit theorem, Markov chains.
4	Texts/References	1. W. Feller, An Introduction to Probability Theory and its Applications, Vol.1, 3rd Edition, John Wiley. 2. G. R. Grimmett and D. R. Stirzaker, Probability and Random Processes, 4th Edition, Oxford University Press, 2020. 3. N. L., Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press, 1989. 4. Jiří Matoušek, Invitation to Discrete Mathematics, 2nd Edition, Oxford University Press, 2008

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Introduction to Programming – 1** **(3-0-2-4)**
2	**Pre-requisite courses(s)**	--
3	**Course content**	This course provides an introduction to problem solving with computers using C Topics covered will include: **Utilization**: Developer fundamentals such as editor, integrated programming environment, Unix shell, modules, libraries. **Programming features**: Machine representation, data types, arrays and records, objects, expressions, control statements, iteration, procedures, functions and recursion, Pointers, Structures and basic I/O. **Applications**: Sample problems in engineering, science, text processing, and numerical methods.
4	**Texts/References**	The C Programming Language Brian W Kernighan, Dennis M Ritchie, Prentice Hall India , 2nd edition, 1988 Programming with C (Second Edition) Byron Gottfried, Schaum's Outlines Series, Tata-Mcgraw Hill, 2011 How to Solve It by Computer, by G. Dromey, Prentice- Hall, Inc., Upper Saddle River, NJ, 1982. How to Solve _It (2nd ed.), by Polya, G., Doubleday and co, 1957. Let Us C, by Yashwant Kanetkar, Allied Publishers, 1998.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Introduction to High Performance Computing** **(0.5-0-0-1)**
2	**Pre-requisite courses(s)**	Nil
3	**Course content**	Introduction to High Performance Computing (HPC): Need for HPC and supercomputing in scientific and engineering applications, definitions, serial and parallel computations Taxonomy of Computer Architectures: Singe Instruction Stream Single Data Stream(SISD),Singe Instruction Stream Multiple Data Stream (SIMD), Multiple Instruction Stream Single Data Stream (MISD), and Multiple Instruction Stream Multiple Data Stream (MIMD) architectures Parallel Computing on parallel Machines Parallel Computing on SIMD, MISD and MIMD architectures; vector and matrix operations Embarrassingly Parallel Applications Monte Carlo methods, fractals and images processing Other Applications Numerical algorithms and applications in science and engineering.
4	**Texts/References**	1. E. Aubanel,'Elements of Parallel Computing', 1st Editon, Chapman and Hall/CRC,2016. 2. A. Grama, G. Karypis, V. Kumar and Gupta, 'An Introduction to Parallel Computing: Design and Analysis of Algorithms', 2nd Edition, Addison Wesley, Reading,2003.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Data Structures and Algorithms** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Computer Programming
3	**Course content**	Introduction: data structures, abstract data types, analysis of algorithms. Creation and manipulation of data structures: arrays, lists, stacks, queues, trees, heaps, hash tables, balanced trees, tries, graphs. Algorithms for sorting and searching, order statistics, depth-first and breadth-first search, shortest paths and minimum spanning tree.
4	**Texts/References**	1. Introduction to Algorithms, 3rd edition, by T. Cormen, C. Leiserson, R. Rivest, C. Stein, MIT Pressand McGraw-Hill, 2009. 2. Data structures and algorithms in C++, by Michael T. Goodrich, Roberto Tamassia, and David M. Mount, Wiley, 2004.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Data Structures and Algorithms Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	Exposure to Computer Programming (CS 102)
3	**Course content**	The laboratory course for CS 211 is based on creatingand manipulating various data structures and implementation of algorithms.
4	**Texts/References**	**1.** Introduction to Algorithms, 3rd edition, by T. Cormen, C. Leiserson, R. Rivest, C. Stein, MIT Pressand McGraw-Hill, 2009. **2.** Data structures and algorithms in C++, by Michael T. Goodrich, Roberto Tamassia, and David M. Mount, Wiley, 2004.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Automata Theory** **(3-1-0-8)**
2	**Pre-requisite courses(s)**	Exposure to Discrete Structures
3	**Course content**	Finite state machines (DFA/NFA/epsilon NFAs), regular expressions. Properties of regular languages. My hill-Nerode Theorem. Non-regularity. Push down automata. Properties of context-free languages. Turing machines:Turing hypothesis, Turing computability, Nondeterministic, multi tape and other versions of Turing machines. Church thesis, recursively enumerable sets and Turing computability. Universal Turing machines. Unsolvability, The halting problem, partial solvability, Turing enumerability, acceptability and decidability, unsolvable problems about Turing Machines. Post`s correspondence problem.
4	**Texts/References**	1. Introduction to Automata Theory, Languages and Computation, by John. E. Hopcroft, Rajeev Motwani, J. D. Ullman, 3rd edition. Pearson. 2013. 2. Elements of the Theory of Computation, by H.R. Lewis and C. H. Papadimitrou, 2nd Edition. Prentice Hall Inc, 1998.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Discrete Structures (3-0-0-6)
2	Pre-requisite courses(s)	--
3	**Course content**	There are four modules in the course: 1. **Proofs and structures** Introduction, propositions, predicates, examples of theorems and proofs, types of proof techniques, Axioms, Mathematical Induction, Well-ordering principle, Strong Induction, Sets, Russell's paradox, infinite sets, functions, Countable and uncountable sets, Cantor's diagonalization technique, Relations, Equivalence relations, partitions of a set. 2. **Counting and Combinatorics** Permutations, combinations, binomial theorem, pigeon hole principle, principles of inclusion and exclusion, double counting. Recurrence relations, solving recurrence relations. 3. **Elements of graph theory** Graph models, representations, connectivity, Euler and Hamiltonian paths, planar graphs, Trees and tree traversals. 4. **Introduction to abstract algebra and number theory** Semigroups, monoids, groups, homomorphisms, normal subgroups, congruence relations. Ceiling, floor functions, divisibility. Modular arithmetic, prime numbers, primality theorems.
4	**Texts/References**	5. Discrete Mathematics and its applications with Combinatorics and graph theory, 7th edition, by Kenneth H Rosen. Special Indian Edition published by McGraw-Hill Education, 2017. 6. Introduction to Graph Theory, 2nd Edition, by Douglas B West. Eastern Economy Edition published by PHI Learning Pvt. Ltd, 2002. 7. Discrete Mathematics, 2nd Edition, by Norman L Biggs. Indian Edition published by Oxford University Press, 2003.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Design and Analysis of Algorithms** (3-0-0-6)
2	**Pre-requisite courses(s)**	Computer Programming and Utilization, Discrete Structures, Data Structures and Algorithms , Data Structures and Algorithms Laboratory
3	**Course content**	Syllabus is divided roughly into 8 modules; each module roughly takes two weeks. **Module 1:** Introduction Examples and motivation. Asymptotic complexity: informal concepts, formal notation, examples **Module 2**: Searching in list: binary search, Sorting: insertion sort, selection sort, merge sort, quicksort, stability and other issues. **Module 3**: Divide and conquer: binary search, recurrence relations. nearest pair of points, merge sort, integer multiplication, matrix multiplication. **Module 4**: Graphs: Motivation, BFS, DFS, DFS numbering and applications, directed acyclic graphs, directed acyclic graphs, Shortest paths: unweighted and weighted, Single source shortest paths: Dijkstra, Minimum cost spanning trees: Prim's algorithm, Kruskal's Algorithm **Module 5**: Union-Find data structure, Priority queues, heaps. Heap sort. Dijstra/Prims revisited using heaps, Search Trees: Introduction Traversals, insertions, deletions Balancing **Module 6**: Greedy algorithms: Greedy: Interval scheduling, Proof strategies, Huffman coding. **Module 7**: Dynamic Programming: weighted interval scheduling, memoization, edit distance, longest ascending subsequence. matrix multiplication, shortest paths: Bellman Ford, shortest paths: Floyd Warshall **Module 8:** Intractability: NP completeness, reductions, examples, Misc topics.
4	**Texts/References**	1. Algorithms, by Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, McGraw Hill Education, 2006. 2. Introduction to Algorithms, 3rd edition, by Cormen, Leiserson, Rivest and Stein, PHI Learning Pvt. Ltd., 2010. 3. Algorithm Design, 1st edition, by Kleniberg and Tardos, Pearson, 2014.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Artificial Intelligence (3-0-0-6)
2	Pre-requisite courses(s)	
3	Course content	Search: Problem representation; State Space Search; A* Algorithm and its Properties; AO* search, Minimax and alpha- beta pruning, AI in games. Logic: Formal Systems; Notion of Proof, Decidability, Soundness, Consistency and Completeness; Predicate Calculus (PC), Resolution Refutation, Herbrand Interpretation, Prolog. Knowledge Representation: PC based Knowledge Representation, Intelligent Question Answering, Semantic Net, Frames, Script, Conceptual Dependency, Ontologies, Basics of Semantic Web. Leaning: Learning from Examples, Decision Trees, Neural Nets, Hidden Markov Models, Reinforcement Learning, Learnability Theory. Uncertainty: Formal and Empirical approaches including Bayesian Theory, Fuzzy Logic, Non-monotonic Logic, Default Reasoning. Planning: Blocks World, STRIPS, Constraint Satisfaction, Basics of Probabilistic Planning. Advanced Topics: Introduction to topics like Computer ain
4	Texts/References	Text: Stuart J. Russel, Peter Norvig, Artificial Intelligence: A Modern Approach (3rd ed.). Upper Saddle River: Prentice Hall, 2010. Other references: N.J. Nilsson, Principles of Artificial Intelligence, Morgan Kaufmann, 1985. Malik Ghallab, Dana Nau, Paolo Traverso, Automated Planning: Theory & Practice, The Morgan Kaufmann Series in Artificial Intelligence, 2004. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006. Mark Stefik, Introduction to Knowledge Systems, Morgan Kaufmann, 1995. E. Rich and K.Knight, Artificial Intelligence, Tata McGraw Hill, 1992.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Software Systems Laboratory** (1-3-0-8)
2	**Pre-requisite courses(s)**	--
3	**Course content**	Vim/emacs HTML, CSS • Report and presentation software: latex, beamer, drawingsoftware(e.g. inkscape, xfig, open-office) • IDE (e.g. eclipse, netbeans), code reading,debugging Basic Java Java collections, interfaces • Java threads Java GUI Introduction to documentation: e.g. doxygen/javadocs • Version management: SVN/Git • Unix basics: shell, file system, permissions, process hierarchy, process monitoring, ssh, rsync • Unix tools: e.g. awk, sed, grep, find, head, tail, tar,cut, sort • Bash scripting: I/O redirection, pipes • Python programming • Makefile, libraries and linking • Graph plotting software (e.g., gnuplot) • Profiling tools (e.g., gprof, prof) • Optional topics (may be specific to individualstudents 302222 projects): intro to sockets, basic SQL for data storage,JDBC/pygresql A project would be included which touches upon manyof the above topics, helping students see the connection across seemingly disparate topics. The project is also expected to be a significant load: 20-30 hours of work.
4	**Texts/References**	**1.** Online tutorials for HTML/CSS, Inkscape, OODrawUnix Man Pages for all unix tools, AdvancedBash Scripting Guide from the Linux Documentation Project (www.tldp.org). **2.** The Python Tutorial Online Book(http://docs.python.org/3/tutorial/index.html). **3.** The Java Tutorials (http://docs.oracle.com/javase/tutorial/). **4.** Latex - A document preparation system, 2/e, byLeslie Lamport, Addison-Wesley, 1994.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Artificial Intelligence Lab** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	
3	**Course content**	The lab will closely follow and aim to elucidate the lessons covered in the theory course CS344. Implementation and study of A*, Usage of Prolog Inferencing, Expert System Shells, Neural Net Platforms, Prediction and Sequence Labeling using HMMs, Simulation of Robot Navigation and such exercises are strongly recommended.
4	**Texts/References**	Stuart J. Russel, Peter Norvig, Artificial Intelligence: A Modern Approach (3rd ed.). Upper Saddle River: Prentice Hall, 2010. Other references: N.J. Nilsson, Principles of Artificial Intelligence, Morgan Kaufmann, 1985. Malik Ghallab, Dana Nau, Paolo Traverso, Automated Planning: Theory & Practice, The Morgan Kaufmann Series in Artificial Intelligence, 2004. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006. Mark Stefik, Introduction to Knowledge Systems, Morgan Kaufmann, 1995. E. Rich and K.Knight, Artificial Intelligence, Tata McGraw Hill, 1992.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Computer Architecture** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	--
3	**Course content**	The Language of Bits, Assembly Language, LogicGates, Registers, and Memories, Processor Design,Principles of Pipelining, The Memory System, Multiprocessor Systems, I/O and Storage Devices. Each concept will be first taught on the basis of the fundamental driving principles. Following this, real world examples (e.g., ARM processors) will be used to emphasize the content.
4	**Texts/References**	1. Computer Organization and Architecture, by SmrutiRanjan Sarangi, McGraw Higher Ed, 2017. 2. Computer Architecture A Quantitative Approach, Sixth edition, by David Patterson and John L. Hennesy,Morgan Kaufmann, 2017.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Data Bases and Information Systems** (3-0-0-6)
2	**Pre-requisite courses(s)**	--
3	**Course content**	Overview of data management systems. Relational model and query languages (relational algebra and calculus, SQL). Database design using the ER Model, ER Diagrams, UML Class Diagrams. Relational database design and normalization. Integrity and Security. Design and development of Web based information systems. Overview of storage structures and indexing, query processing and optimization, and transaction processing. Introduction to Big Data management concepts such as: distributed and scalabledata storage, including distributed file systems, keyvalue stores, column stores and graph databases, replication and consistency, and concurrent data processing using the Map Reduce paradigm. Introduction to decision support and data analysis, data warehousing and data mining, and Information Retrieval.
4	**Texts/References**	Database System Concepts, 6th edition, by AbrahamSilberschatz, Henry F. Korth and S. Sudarshan, McGraw Hill, 2010.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Operating Systems** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Computer Architecture
3	**Course content**	Process Management, Memory Management, Storage Management, Protection and Security, Virtual Machines, Distributed Systems
4	**Texts/References**	Avi Silberschatz, Peter Baer Galvin, Greg Gagne, ``Operating Systems Concepts" 9th edition. *Wiley*. Andrew S. Tanenbaum, Herbert Bos, ``Modern Operating Systems", 4th edition. *Pearson*

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Introduction to Artificial Neural Networks** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Nil
3	**Course content**	Background to ANN and PDP models; Basics of ANN including terminology, topology and learning laws; (4 lectures) Analysis of Feedforward neural networks (FFNN) including linear associative networks, perceptron network, multilayer perceptron, gradient descent methods and backpropagation learning; (8 lectures) Analysis of Feedback neural networks (FBNN) including Hopfield model, state transition diagram, stochastic networks, Boltzmann learning law; (8 lectures) Evolution of ANN architectures - from learning to deep learning: (1 lecture)
4	**Texts/References**	**1.** B Yegnanarayana, Artificial Neural Networks, Prentice Hall of India, New Delhi, 1999. **2.** David E Rumelhart, James L McClelland, and the PDP Research group, Eds, Parallel and Distributed Processing: Explorations in Microstructure of Cognition, Vol.1, Cambridge MA: MIT Press, 1986a **3.** James L McClelland, David E Rumelhart and the PDP Research group, Eds, Parallel and Distributed Processing: Explorations in Microstructure of Cognition, Vol.2, Cambridge MA: MIT Press, 1986b **4.** James L McClelland, David E Rumelhart and the PDP group, Eds, Explorations in Parallel and Distributed Processing: A Handbook of Models, Cambridge MA: MIT Press, 1989 **5.** Simon Haykin, Neural Networks and Learning Machines, Pearson Education, 2011 **6.** Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep learning, MIT Press, 2017.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Computer Architecture Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	--
3	**Course content**	The lab will closely follow the theory course. The idea is to have the students develop a software model of a simple processor, capturing both functionality and timing aspects. They will implement modules as the concepts are taught in class.
4	**Texts/References**	Nil

Computer Science Engineering

1	Title of the course (L-T-P-C)	Data Bases and Information Systems Laboratory (0-0-3-3)
2	Pre-requisite courses(s)	--
3	Course content	Use of database systems supporting interactive SQL. Two-tier client-server applications using JDBC or ODBC, Three-tier web applications using Java servlets/JDBC orequivalent. Design of applications and user interfaces using these systems. Data analysis tools. Laboratory project involving building data backed applications withWeb or mobile app frontends.
4	Texts/References	Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts 6th Ed, McGrawHill, 2010.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Operating Systems Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	**Computer Architecture**
3	**Course content**	Laboratory Assignments related to the topics covered in the theory course: Process Management, Memory Management, Storage Management, Protection and Security, Virtual Machines, Distributed Systems
4	**Texts/References**	1. **Avi Silberschatz, Peter Baer Galvin, Greg Gagne**, ``Operating Systems Concepts'' 9th edition. Wiley. 2. **Andrew S. Tanenbaum, Herbert Bos**, ``Modern Operating Systems", 4th edition. Pearson.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Computer Networks Laboratory** **(0-0-3-3)**
2	**Pre-requisite courses(s)**	Nil
3	**Course content**	Experiments to support study of the Internet protocol stack: (a) Experimental study of application protocols such as HTTP, FTP, SMTP, using network packet sniffers and analyzers such as Ethereal. Small exercises in socket programming in C/C++/Java. (b) Experiments with packet sniffers to study the TCP protocol. Using OS (netstat, etc) tools to understand TCP protocol FSM, retransmission timer behavior, congestion control behaviour. (c) Introduction to ns2 (network simulator) - small simulation exercises to study TCP behavior under different scenarios. (d) Setting up a small IP network - configure interfaces, IP addresses and routing protocols to set up a small IP network. Study dynamic behaviour using packet sniffers (e) Experiments with ns2 to study behaviour (especially performance of) link layer protocols such as Ethernet and 802.11 wireless LAN.
4	**Texts/References**	Nil

Computer Science Engineering

1	Title of the course (L-T-P-C)	Compilers Lab (0-0-3-3)
2	Pre-requisite courses(s)	Exposure to Data Structures and Algorithms, Computer Architecture, Automata Theory, and a programming language such as C/C++/Java.
3	Course content	Design and implementation of a scanner using scanner generator. Design and implementation of a parser using parser generator. Symbol table generation, Semantic actions for expressions, control structures, and functions. Implementing liveness analysis and applying it to register allocation.
4	Texts/References	1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D.Ullman: Compilers: Principles, Techniques, and Tools, 2/E, AddisonWesley 2007. 2. Andrew Appel: Modern Compiler Implementation in C/ML/Java, Cambridge University Press, 2004 3. Dick Grune, Henri E. Bal, Cerial J.H. Jacobs and Koen G. Langendoen: Modern Compiler Design, John Wiley & Sons, Inc. 2000. 4. Michael L. Scott: Programming Language Pragmatics, Morgan Kaufman Publishers, 2006. 5. Fisher and LeBlanc: Crafting a Compiler in C.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Compilers (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to Data Structures and Algorithms, Computer Architecture, Automata Theory
3	Course content	The compiled and interpreted execution models. Lexical analysis and parsing using lex and yacc. LR parsers, Scope and visibility analysis. Data layout and lifetime management of data. Runtime environment. Dynamic memory allocation and Garbage collection. Translation of expressions, control structures, and functions. Code generation and introduction to optimizations (local and global). Lattice Theory, Optimizations- dataflow, control flow, reaching definition, liveness analysis, code transformation-tiling, fusion.
4	Texts/References	1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D.Ullman: Compilers: Principles, Techniques, and Tools, 2/E, AddisonWesley 2007. 2. Andrew Appel: Modern Compiler Implementation in C/ML/Java, Cambridge University Press, 2004 3. Dick Grune, Henri E. Bal, Cerial J.H. Jacobs and Koen G. Langendoen: Modern Compiler Design, John Wiley & Sons, Inc. 2000. Michael L. Scott: Programming Language Pragmatics, Morgan Kaufman Publishers, 2006. 4. Fisher and LeBlanc: Crafting a Compiler inC.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Programming Techniques** **(3-0-0-3)**
2	**Pre-requisite courses(s)**	A first-level course in any programming language.

3	**Course content**	**The programming Techniques** course starts with refresher topics on programming language constructs and data structures. Then, the discussion switches to data structures and algorithms for problem solving. The course is meant for students who are familiar with a programming language (preferably C++ and Python) and those who desire to improve techniques of problem solving and programming.
		 1. Programming Refresher
		 2. Data structures refresher, ADTs
		    1. Arrays, lists, multilists, queues, and stacks.
		 3. Non-linear data structures
		    1. Trees, Graphs, Hash Tables, Tries, Suffix trees
		 4. Algorithms
		    1. Basic analysis, complexity
		    2. Design techniques
		 5. Some common problems and program design strategies

4	**Texts/References**	1. Course Notes 2. How to Solve it by computers: Dromey, Pearson, 2006 3. The C Programming Language: Kernighan and Ritchie, Second Edition, Pearson, 2015 4. Data Structures and program design: Kruse, Pearson, 2006 5. Introduction to the Design and Analysis of Algorithms, Anany Levitin, third edition, 2017 6. Programming Pearls, Jon Bentley, 2002.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Computer Networks** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Nil
3	**Course content**	Design of Computer Networking protocols at all layers: transmission media, data link protocols, media access control, routing and congestion control, admission control, traffic shaping and policing, Internet working (IP) and transport layer protocols (TCP). Performance analysis of networks.
4	**Texts/References**	**1.** Data and Computer Communications, 6th edition, by W. Stallings, Prentice Hall, 2000. **2.** Computer Networks, 4th edition, by A. S. Tannenbaum, Prentice Hall, 2003. **3.** Data Communications, Computer Networks and Open Systems, 4th edition, by F. Halsall, Addison-Wesley, 1996. **4.** High Performance Communication Networks, by Walrand and Varaiya, Morgan Kaufman, 1996. **5.** Internet working with TCP/IP: Principles, Protocols, Architecture, 3rd edition, by D. E. Comer, Prentice Hall, 1996. **6.** TCP/IP Illustrated Vol. I, by W. R. Stevens, Addison Wesley, 1994.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Software Engineering (3-0-0-6)
2	Pre-requisite courses(s)	
3	Course content	**Introduction** What is Software Engineering. **Software Development Lifecycle** Requirements analysis, software design, coding, testing, maintenance, etc. **Software life-cycle models** Waterfall model, prototyping, interactive enhancement, spiral model. Role of Management in software development. Role of metrics and measurement. **Software Requirement Specification** Problem analysis, requirement specification, validation, metrics, monitoring and control. **System Design** Problem partitioning, abstraction, top-down and bottom-up design, Structured approach. Functional versus object-oriented approach, design specification and verification metrics, monitoring and control. Software Architecture **Coding** Top-down and bottom-up, structured programming, information hiding, programming style, and internal documentation. Verification, Metrics, monitoring and control. **Testing** Levels of testing functional testing, structural testing, test plane, test cases specification, reliability assessment. **Software Project Management** Cost estimation, Project scheduling, Staffing, Software configuration management, Quality assurance, Project Monitoring, Risk management, etc. including tools for software development to release, supporting the whole life cycle.
4	Texts/References	1. Software Engineering: A Practioner's approach, R.S. Pressman, McGraw Hill, 8th edition. 2. Introduction to Software Engineering, Pankaj Jalote, Narosha Publishing 3. The Unified Software Development Process, I. Jacobson, G. Booch, J. Rumbaugh, Pearson Education 4. Software Architecture in Practice, L. Bass, P. Clements, R. Kazmann, 3rd ed., Addison Wesley

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Distributed Systems** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Operating Systems, Data Structures and Algorithms, Programming in C++
3	**Course content**	- Introduction to distributed systems, Message Passing, Leader Election, Distributed Models, Causality and Logical Time - Logical Time, Global State & Snapshot and Distributed Mutual Exclusion-Non- Token and Quorum based approaches - Distributed Mutual Exclusion-Token based approaches, Consensus & Agreement, Checkpointing & Rollback Recovery - Deadlock Detection, DSM and Distributed MST - Termination Detection, Message Ordering & Group Communication, Fault Tolerance and Self-Stabilization, Gossip Style communication, chord, pastry - Concurrency and Replication Control, RPCs, Transactions - Distributed Randomized Algorithms, DHT and P2P Computing - Case Studies: GFS, HDFS, Map Reduce and Spark
4	**Texts/References**	1. Distributed Computing: Principles, Algorithms, and Systems- Ajay D. Kshemkalyani and Mukesh Singhal 2. Distributed Computing: Fundamentals, Simulations and Advanced Topics-Hagit Attiya and Jennifer Welch 3. Distributed Algorithms-Nancy Lynch 4. Elements of Distributed Computing-Vijay K. Garg 5. Advanced Concepts in Operating Systems-Mukesh Singhal, Niranjan G. Shivaratri

Computer Science Engineering

1	Title of the course (L-T-P-C)	Graph Theory and Combinatorics (3-0-0-6)
2	Pre-requisite courses(s)	Discrete Structures
3	Course content	Fundamentals of graph theory. Topics include: connectivity, planarity, perfect graphs, coloring, matchings and extremal problems. Basic concepts in combinatorics. Topics include: counting techniques, inclusion-exclusion principles, permutations, combinations and pigeon- hole principle.
4	Texts/References	"An Introduction to Quantum Field Theory", Michael Peskin and Daniel Schroeder (Addison Wesley) "Introduction to Quantum Field Theory", A. Zee "Quantum Field Theory", Lewis H. Ryder "Quantum Field Theory and Critical Phenomena", by Jean Zinn-Justin. "Quantum field Theory for the Gifted Amateur", T. Lancaster and Stephen J. Blundell NPTEL lectures in Quantum Field Theory (https://nptel.ac.in/courses/115106065/)

Computer Science Engineering

1	Title of the course (L-T-P-C)	Parallel Computing (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to C, C++ or Fortran programming
3	Course content	Need for High Performance Computing (HPC) and applications. Sequential Computing model, Algorithms and their complexity. Taxonomy of computer architectures – SISD, SIMD (e.g. array processors), MISD (pipelined processing, vector processors), and MIMD (shared memory and distributed memory multiprocessors, computing clusters); dataflow computing; hardware accelerators (GPUs); interconnection networks (bus, loop, mesh and hypercube); Memory hierarchy; Case Studies. Implications of computer architectures to algorithm design, synchronous processing, single program multiple data (SPMD) and multiple program multiple data (MPMD) processing; functional and data parallelism; memory hierarchies. Performance evaluation: communication and computing costs, speedup, efficiency, Amdahl's law, parallel scalability. Parallel algorithm design and case studies: numerical algorithms (linear algebra, matrix- vector and matrix-matrix multiplications, finite difference method and PDEs, Monte Carlo method), and non-numerical algorithms (search, sorting, simple tree and graph algorithms) Parallel programming platforms, OpenMP and MPI programming, GPU programming. Programing Assignments: 1. Parallel computing lab environment (system architecture, log on, hello world 2. Editors, job submission, optimization techniques for serial code. 3. MPI and simple program(s) 4. MPI and matrix-matrix multiplication 5. OpenMP and matrix-matrix multiplication OpenMP 6. Introduction to GPU programming – matrix-matrix multiplication.
4	Texts/References	1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: Introduction to Parallel Computing, Addison Wesley 2003 2. Eric Aubanel, Elements of Parallel Computing, CRC Press, 2017. 3. https://computing.llnl.gov/tutorials/mpi/ 4. https://computing.llnl.gov/tutorials/open MP/

Computer Science Engineering

1	Title of the course (L-T-P-C)	Logic for Computer Science (3-0-0-6)
2	Pre-requisite courses(s)	Fundamental concepts and applications of chemistry (CH101)
3	Course content	Atomic Bonding, Crystal Structure and Defects, Mechanical and Thermal Behaviour, Metals and alloys, Semiconductors, Ceramics & Glasses, Mechanical properties of inorganic and composite materials. Biomaterials. 1. Basic characteristics of polymers, glass transition temperature, crystallinity, molecular weight and molecular weight distribution. Various types of Polymer Synthesis characterization; Chemical analysis of polymers; Conducting polymers and their applications in optoelectronics and sensors.
4	Texts/References	1. Introduction to Materials Science for Engineers; 8th Edition James F. Shackelford 696 pages; Pearson (April 12, 2014) 2. Materials Science and Engineering: An Introduction; 9th Edition William D. Callister Jr.and David G. Rethwisch 984 pages; Wiley (December 4, 2013) 3. M. Campbell, Introduction to Synthetic Polymers, 2nd Ed, Oxford University Press, New York (2000) 4. Principles of polymerization: G. Odian, 4th Edition, John Wileyinterscience 5. Conducting polymers, fundamentals and applications: A practical approach, Prasanna Chandrasekhar, Springer Science+Business media

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Principles of Programming Languages** (3-0-0-6)
2	**Pre-requisite courses(s)**	Knowledge of a programming language, preferably multiple multi-paradigm programming languages. Mathematical and analytical inclination.
3	**Course content**	**Foundations of Programming Languagses** is an undergraduate introductory course that covers formal aspects of computations and programming language constructs. While the course discusses formal aspects, it also draws from different programming languages like Haskell, ML, Clojure, Java, Scala and C++. Some of the topics that will be covered are: **1.** Types and introduction to type theory **2.** Recursive Types **3.** Untyped and typed lambda calculus **4.** Types and Subtypes and polymorphism **5.** Object Calculus **6.** Concurrent Programming **7.** Reflective Programming
4	**Texts/References**	Text: **1.** Essentials of Programming Languages, Friedman, Wand and Haynes, PHI, 2012. **2.** Foundations of Object-Oriented Languages, Kim Bruce, PHI 2004. Papers from literature

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Elements of Programming** (2-0-2-6)
2	**Pre-requisite courses(s)**	None
3	**Course content**	**Elements of Programming** is a first-level course in programming. Starting with the basics of programming, it culminates in fairly complex programs covering program design, structure and techniques along the way. This course uses Python programming language to learn the principles. Topics covered include: 1. Program structure 2. Types and operations on types 3. Abstractions in programs 4. Object-Oriented Programming 5. Functional Programming 6. Exceptions 7. Program performance 8. Problem Solving techniques – data structures and algorithms 9. Programming for data analytics
4	**Texts/References**	1. Required Text: Introduction to Computation and Programming Using Python, 2nd Edition, by John V. Guttag, Prentice Hall of India, 2016 2. Reference: 1. Think Like a Programmer, by V. Anton Spraul, No Statch Press, 2012

Computer Science Engineering

1	Title of the course (L-T-P-C)	Introduction to Blockchains (3-0-0-6)
2	Pre-requisite courses(s)	It is expected that students should have a very good programming background preferably in Java/JavaScript and Python.
3	Course content	Basics - What is Blockchain, Centralized vs Decentralized vs Distributed, Blockchains and Public Ledgers, Fundamentals - Cryptographically secure hash functions, Merkle Trees, Bitcoin Concept Architecture - Blockchain 2.0, smart contracts, block structure, notion of distributed consensus, challenge response to Permission-less Consensus, economics behind blockchain consensus Consensus in Blockchain - Distributed Consensus, Proof of Work, Miners in the context of Bitcoin Permissioned Blockchain - Basics, RAFT Consensus, Byzantine General Problem, Practical Byzantine Fault Tolerance Hyperledger Fabric - Introduction, Transaction flow, Membership and Identity Management, Hyperledger Composer" Ethereum Framework - Installation and subsequent execution of the use cases. Blockchain in Financial Service - Payments and Secure Trading, Compliance and Mortgage, Financial Trade Blockchain in Supply Chain Blockchain in Government Use Cases - Digital Identity" Mini Project implementation using Ethereum/Hyperledger framework on use cases related to financial, supply chain, and government sectors
4	Texts/References	Arvind Narayanan, Joseph Bonneu, Edward Felten, Andrew Miller, Steven Goldfeder, "Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction", Princeton University Press, 2016. Research papers as well as several related online educational content will also be referred to for learning some of the afore-mentioned topics.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Mathematics for Data Science (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to basic concepts in calculus and linear algebra
3	Course content	Introduction to Data science and Motivation for the course. Review of calculus, naTve set theory, notion of limits, ordering, series and its convergence. Introduction to Linear Algebra in Data science, notion of vector space, dimension and rank, algorithms for solving linear equations, importance of norms and notion of convergence, matrix decompositions and its use. Importance of optimization in data science: Birds view of Linear Regression, Multivariate Regression, Logistic Regression etc. Convex Optimization: Convex sets, convex functions, duality theory, different types of optimization problems, Introduction to linear program. Algorithms: Central of gravity method, Gradient descent methods, Nestrov acceleration, mirror descent/ Nestrov dual averaging, stochastic gradient methods, Rmsprop, SIGNSGD, ADAM algorithm etc. Non-convex optimization: Demonstration of convex methods on non- convex problems; merits and disadvantages.
4	Texts/References	1. C. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006. 2. Cambridge university press, 2018 (reprint). for Machine Learning," Now publisher, 2017.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Deep Learning (3-0-0-6)
2	Pre-requisite courses(s)	Exposure to Calculus, Linear Algebra, Probability, Random Processes, Ability to code in Python
3	Course content	**Introductory Concepts of DNN** 1. Linear regression, logistic regression –penultimate layers of a neural network 2. Dealing with nonlinearity – Kernal Trick 3. Data-driven kernel learning using NNs **DNN Training** 1. Issues in training practical deep networks, Vanishing/Exploding gradients 2. Regularization for Deep Learning – Early stopping, weight regularization, activity regularization, dropout 3. Optimization methods for training deep networks – Stochastic gradient descent, rmsprop, adam 4. Convolutional Neural networks **Sequence Modeling** 1. Recurrent neural network 2. LSTMs ans BLSTMs **Unsupervised Learning** 1. Autoendcoder 2. Variational autoencoder 3. Generative adversial networks (GANs) 4. Representation learning and feature extraction.
4	Texts/References	1. Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning," MIT Press 2. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press. 1995 3. B Yegnanarayana, "Artificial Neural Networks," PHI.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Computer Graphics (3-0-2-8)
2	Pre-requisite courses(s)	Exposure to Computer programming and basic linear algebra.
3	Course content	Introduction to the course, Rasterization Basics, Drawing in OpenGL, Clipping 2D Transformations, 3D Transformations, Viewing Transformations, The Modeling-Viewing Pipeline, Visibility, Hierarchical Modelling, Shading, Texture Cubic Splines, Bezier Splines, B-Splines Principles of Animation, Interpolation for Animation, Modelling Surfaces
4	Texts/References	**1.** Fundamentals of Computer Graphics (Third Edition), Peter Shirley, Steve Marschner and others, A K Peters/CRC Press (2009) **2.** Interactive Computer Graphics - A Top- Down Approach Using OpenGL (6/e), Edward Angel **3.** Computer Graphics using OpenGL (3/e), F. S.Hill Jr. and S. M. Kelley **4.** Computer Graphics with OpenGL (3/e), D. D. Hearn and M. P. Baker

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Introduction to Logic** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to Discrete Mathematics
3	**Course content**	**1.** Propositional logic: Natural deduction, Semantics of propositional logic, Soundness of propositional logic, Completeness of propositional logic Horn clauses and satisfiability. **2.** Predicate logic: Natural deduction rules, decidability of predicate logic, Expressiveness of predicate logic **3.** Program correctness: Hoare triples, Partial and total correctness, Proof calculus for partial correctness, Proof calculus for total correctness **4.** Other Applications such as Logic in databases, Logic programming, Puzzle solving 5 Practice with Verification tools.
4	**Texts/References**	**1.** Logic in Computer Science. Huth and Ryan. Cambridge University Press, 2004 **2.** A Mathematical Introduction to Logic. Herbert D Enderton. Harcourt Academic Press. **3.** First-Order Logic and Automated Theorem Proving by Melvin Fitting.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Cloud Software Development (1.5-0-0-3)
2	Pre-requisite courses(s)	**Desirable**: Exposure on Operating System, Database, Cloud Programming language (Java, .Net, NodeJS, HTML/CSS, etc.)
3	Course content	**Module 1 - Introduction to Cloud Computing Landscape** ● Understand how industries rely on the cloud computing global infrastructure, Identify the applications and use cases ● Identify the principles and characteristics of Cloud Computing - IaaS, PaaS, SaaS ● Validate the different patterns of cloud computing adoption including public cloud services, private and hybrid approaches ● Identify common challenges associated with the adoption of cloud computing solutions and associated myths ● Compare and contrast with on-premise/traditional versus cloud ● Understand in-country data regulations, data sovereignty considerations **Module 2 - Cloud Computing Technology** ● Understand Virtualization Concepts - data, compute, network, operating system, HCI ● Understand Cloud Infrastructure -Backup, Restore, Migration, DC/DR, HA use cases ● Understand Programming concepts Cloud-native apps, Serverless, Containers ● Learn Containers– Kubernetes, Docker, containers **Module 3 - Using Managed Cloud Services** ● Learn 12-factor Application Architecture, api, Microservices, databases - sql, no-sql, object store ● Application and Microservice Security- OAuth, access tokens ● Understand Autoscale - horizontal and vertical scaling, logging and monitoring aspects of apps and infrastructure ● Learning DevOps frameworks - toolchains, ci/cd, blue/green deployment, canary deployment **Module 4 - Case Studies - Public Cloud Provider – aws, azure, ibmcloud**
4	Texts/References	Text Books: - Thomas Erl, Zaigham Mahmood, Ricardo Puttini, "Cloud Computing Concepts, Technology & Architecture", Pearson, 2013. Reference Books: - Boris Scholl, Trent Swanson, Peter Jausovec, "Cloud Native", O'Reilly, 2019. Resources from Internet: Public Cloud Documentations: https//learning.oreilly.com/library/view/cloud- computing-concepts/9780133387568/ https://www.amazon.in/Cloud-Computing-Concepts-Technology-Architecture/dp/0133387526/ Class Notes/Lectures

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Programming Parallel Machines** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Good programming experience in C, C++ or Fortran.
3	**Course content**	Elements of parallel algorithm design (domain decomposition, mapping), Parallel hardware architecture - overview of shared-memory systems, cache coherence, and sequential consistency. Shared- memory programming (OpenMP, PThread, Cilk). Scalable algorithm design and measuring performance (speedup, efficiency). Distributed-memory programming (MPI and message-passing model, global address space model). Interconnects. Introduction to GPU programming
4	**Texts/References**	1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: Introduction to Parallel Computing, Addison Wesley 2003. 2. https://computing.llnl.gov/tutorials/mpi/ 3. https://computing.llnl.gov/tutorials/open MP/

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Introduction to Abstractions and Paradigms for Programming (1-0-0-2)**
2	**Pre-requisite courses(s)**	
3	**Course content**	Ideas behind imperative, applicative, object oriented and logic programming paradigms such as typing, expressions, pure functions, recursion, higher order functions, encapsulation, inheritance, goal satisfaction, backtracking, unification.
4	**Texts/References**	Harold Abelson, Gerald Jay Sussman and July Sussman, Structure and Interpretation of Computer Programs, 2nd edition, The MIT Press, 1996. David A. Watt, Programming Language Concepts and Paradigms, Prentice-Hall, 1990. Rajeev Sangal, Programming Paradigms in Lisp, McGraw Hill, 1991.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Operational Analysis** **(3-0-0-6)**
2	**Pre-requisite courses(s)**	Exposure to programming languages (C/C++/MATLAB), ability to rig up basic electrical/electronic circuits
3	**Course content**	Part 1: Stages of manufacturing supply chain and value addition. Concepts of supply, storage, production, warehousing and transport. Supply chain terminology. Structure of supply chains, decisions and decision levels in supply chain. Overview of supply chain operations & costs: location, procurement, production, inventory, transportation, and information technology. Material flow systems: push, pull, flow shop, job shop, cell, etc. Part 2: Quantitative models in supply chain systems including forecasting, production planning, scheduling and inventory models. Basic forecasting models. Product structure representation. Bill of materials. Material accounting logic and MRP systems. Scheduling and sequencing of parts: single/two machine. Notion of quality & quality control. Quality and yield of manufacturing processes. Inventory models: EOQ, periodic review, continuous review. Introduction to procurement and distribution models. Transport & logistics costs. Project management techniques: CPM and PERT.
4	**Texts/References**	1. Steven Nahmias (2004) Production and Operations Analysis, 5th edition, McGraw-Hill. 2. W. Hopp and M. Spearman (2000) Factory Physics, 3rd edition, McGraw-Hill. 3. Sunil Chopra and Peter Meindl (2006) Supply Chain Management, 3rd edition, Pearson Education India. 4. Joseph S. Martinich (2000) Production and Operations Management: An Applied Modern Approach, John Wiley & Sons.

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Advanced in Cloud Technologies**
2	**Pre-requisite courses(s)**	-

3	**Course content**	**Chapter-1: Introduction to cloud (2 Hours)**
		a. Public
		b. Private
		c. On-Prem
		d. Managed cloud services: IaaS/PaaS/Saas
		e. Architectures for the cloud software
		    i. Monoliths Vs Microservers
		    ii. Serverless Vs Servers
		f. Cloud nativity:
		    i. Stateful Vs Stateless
		    ii. Platform independence and scale (designing with chaos in mind)- https://12factor.net/
		    iii. Infrastructure-as-code
		    iv. DevOps-(clous-native CI/CD)
		    v. Multi-cloud and Hybrid-cloud
		
		Chapter 2: Virtualization(4 Hours)
		a. What is Virtualisation? (Bare metal Servers)
		b. Why virtualisastion?
		c. Virtualization technologies:
		    i. Virtal Machines
		    Hypervisors
		    Practical Illustration: Iibvirt API
		    ii. Container
		    Container Image Layering architecture
		    O Copy-on-write(e.g., Union File System)
		    O namespaces vs. cgroups for resource access
		    Hands-On: Docker/Podman Usage tutorials
		    O Dockerfiles
		    ODocker/Podman commands
		    iii. Emerging technologies
		    Unikernels-some demo illustrations
		    Web Assembly- some demo illustrations
		    Secure virtualization
		    o Secure containers - gVisor (https://gvisor.dev/)
		    o Kata (https://ubuntu.com/kubernetes/docs/kata)
		    o Nabla (https://nabla-containers.github.io/)
		    MicroVM- Firecracker
		
		Chapter3: Orchestration Pltforms (4 hours)
		a. Why we need orchestration?
		b. Popular or orchestration frameworks
		    i. Docker swarm
		    ii. Kubernetes
		        1. Declarative approach
		        2. Abstractions and Resources
		        3. Auto-Scaling
		        4. Flavours of K8s
		        oK8s for thr edge-K3s, KubeEdge, MicroK8s, Single-node OCP
		        o virtual Kubelet
		        5. Multi-cluster
		        O Hypershift
		        O KCP- Kubernetes control plane-creating virtual clusters
		        7. Hands-On
		        O Resource Yamis – pods, deployments, service, ingress, jobs, ststefeulset & headless service, volumes, configmaps and secrets etc.

Computer Science Engineering

<table>
<tr><td></td><td></td><td>
O Helmchats
  i.   Serverless: scale-to-zero
        1.   Functionas-a-service (FaaS)
        2.   Function-as-a-service (FaaS)

Chapter-4: Modernisation (3 Hours)
    a.   Assessment and analysis (e.g.., Konveyor Move2Kude)
    b.   Right-sizing of target cluster
    c.   Approaches to moderanization
     i. Lift-and-shift
       1. Kubevirt:
         https://www.youtube.com/watch?v=-w4Afj5-0_g
          [https://kubevirt.io/]
     ii. Re-platform: Platform modernization
       1. What?
       2. Why?
       3. Challenges
       4. Real-world tool: IBM Move2Kube
         1.   Demo
         2.   Tutorials
iii. Refactor: Application modernization
       1. What?
       2. Why?
       3. Challenges
       4. Real-world tool: IBM Mono2Micro
         1.   Demo
         2.   Tutorials

Chapter 5: Operations (1 Hour)
    a. DevSecOps - Compliance, Security and Governance
    b. Application Resource Management (e.g., IBM Turbonomic)
    Observability (e.g., IBM Instana)
</td></tr>
<tr><td>4</td><td>**Texts/References**</td><td>

1. Konveyor Suite Of Tools: https://konveyor.github.io/
2. Jonas, Eric, Johann Schleier-Smith, Vikram Sreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar et al. "Cloud programming simplified: A berkeley view on serverless computing." *arXiv preprint arXiv:1902.03383* (2019).
3. "Kubernetes Patterns". https://www.oreilly.com/library/view/kubernetes-patterns/9781 492050278/
4. Seshadri, Padmanabha V., Harikrishnan Balagopal, Akash Nayak, Ashok Pon Kumar, and Pablo Loyola. "Konveyor Move2Kube: A Framework For Automated Application Replatforming." In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pp. 115-124. IEEE, 2022.
5. Talks on Move2Kube

[2020-2022]:https://www.youtube.com/playlist?list=PL4aUFFbk56EMwb 9xoYs-3RjgkD38C6nPO

Security as code,

Published 2023:https://learning.oreilly.com/library/view/security-as-code/97810981 27459/

6. Cloud computing: Methodology, Systems and Applications, Published 2017 https://learning.oreilly.com/library/view/cloud-computing/97814398 56420/
7. What is Kubevirt? Orchestrating VMs in Kubernetes, Published 2022:https://learning.oreilly.com/library/view/what-is-kubevirt/978109813 3429/
8. Infrastructure as code, 2nd edition, Published 2020:https://learning.oreilly.com/library/view/infrastructure-as-code/9781 098114664/
9. "Operating Systems: Three Easy Pieces". https://pages.cs.wisc.edu/~remzi/OSTEP/
</td></tr>
</table>

10. "Professional Linux Kernel Architecture". https://www.oreilly.com/library/view/professional-linux-kern el/9780470343432/
12. Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee et al. *Above the clouds: A berkeley view of cloud computing*. Vol. 17. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
13. https://www.udemy.com/course/welcome-to-cloud-computing-world/ (Cloud Computing Fundamentals)
14. https://www.udemy.com/course/cloud-computing-for-beginners-infrastructure-as-a-service/ (Cloud Computing for Beginners - Infrastructure as a Service)
15. https://www.coursera.org/learn/introduction-to-cloud (Introduction to Cloud Computing)
16. https://www.coursera.org/learn/cloud-computing-foundations-duke (Cloud Computing Foundations)
17. https://www.coursera.org/specializations/cloud-computing (Cloud Computing Specialization)
18. https://www.coursera.org/learn/cloud-computing (Cloud Computing Concepts, Part 1)
19. https://www.coursera.org/learn/cloud-computing-2 (Cloud Computing Concepts: Part 2)
20. https://www.coursera.org/learn/fundamentals-of-cloud-computing (Fundamentals of Cloud Computing)
21. https://www.coursera.org/specializations/cloud#courses (System Issues in Cloud Computing Specialization)
22. https://www.coursera.org/learn/codio-cloud-computing-primer-semi-tech business-software-as-a-service-saas
    23. https://www.coursera.org/learn/codio-cloud-computing-primer-semi-tech business-infrastructure-as-a-service-iaas
https://www.coursera.org/learn/codio-cloud-computing-primer-semi-tech business-platform-as-a-service-paas

Computer Science Engineering

1	Title of the course (L-T-P-C)	**Advanced Topics in Deep learning** **(3-0-2-8)**
2	Pre-requisite courses(s)	1. Deep Learning (CS 428) or NNDL (EE620) or equivalent and
3	**Course content**	**Module 1**:**Overview of Deep Learning** Historical Context, Basics and Milestones, Basic Optimization, Convex vs. Non-Convex Optimization, Convergence Rates and Acceleration effects of algorithms, Learning vs. Pure Optimization **Module 2: Optimization Methods and Regularization** Advanced Optimization Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates, Approximate Second Order Methods, Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Early Stopping, Dropout, Tangent Distance **Module 3: Learning Paradigms for Deep Learning** Generalization Properties, Efficient Learning, Guarantees on Learning, Teacher-student network, Zero-Shot Learning, Few-Shot Learning, Orthogonal Learning. **Module 4: Transformer Networks** Limitation of Recurrent Neural Networks (RNNs), Sequence-to-sequence modeling and attention, Transformers, Large language models (LLM) – Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-Trained Transformers (GPT), Transformer for vision and speech applications **Module 5: Advanced Convolutional Networks** Advanced Topics on Transfer Learning, Advanced CNN Architectures and Applications including Object Detection and Recognition, Segmentation (Triplet loss, contrastive loss, and ranking loss), CNN for graph data Module 6: Recent Trends in Deep learning Deep Generative AI - Generative Adversarial Networks (GANs) Series, Evolutionary Deep Learning, Deep Reinforcement Learning, Security and Privacy Concerns in Deep Learning **Module 7**: **Advanced Emerging Models** Attention and Diffusion Models, Physics-Informed Neural Networks, Kolmogorov-Arnold Networks (KAN), Conformers The lab component will closely follow the theory course, allowing students to implement advanced algorithms on different topics studied. Practical sessions will include hands-on experiments with advanced deep learning algorithms through Python programming, covering areas like optimization, regularization techniques, Zeroshot / Fewshot Learning, RNNs, transformers, Advanced CNN Architectures and Applications, Generative AI, Physics informed neural networks, and emerging models.

Computer Science Engineering

4	Texts/References	- Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. I. Vasilev, Advanced Deep Learning with Python: Design and Implement Advanced Next-Generation AI Solutions Using TensorFlow and PyTorch. Packt Publishing Ltd, 2019. Y. Bengio, "Learning deep architectures for AI," Foundations and Trends® in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," Communications of the ACM, vol. 64, no. 3, pp. 107-115, 2021. B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro, "Geometry of optimization and implicit regularization in deep learning," arXiv preprint arXiv:1705.03071, 2017. C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in International Conference on Machine Learning, 2017, pp. 1724-1732. S. Cameron, A. Pretorius, and S. Roberts, "Nonparametric Boundary Geometry in Physics Informed Deep Learning," Advances in Neural Information Processing Systems, vol. 36, 2024. H. Wang, S. Ma, L. Dong, S. Huang, D. Zhang, and F. Wei, "Deepnet: Scaling transformers to 1,000 layers," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," ACM Computing Surveys, vol. 56, no. 2, pp. 1-40, 2023. W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, and J. R. Wen, "A survey of large language models," arXiv preprint arXiv:2303.18223. 2023. N. Li, L. Ma, G. Yu, B. Xue, M. Zhang, and Y. Jin, "Survey on evolutionary deep learning: Principles, algorithms, applications, and open issues," ACM Computing Surveys, vol. 56, no. 2, pp. 1-34, 2023.

Computer Science Engineering

1	**Title of the course** (L-T-P-C)	**Adversarial Machine Learning** (3-0-0-6)
2	**Pre-requisite courses(s)**	Nil
3	**Course content**	**Unit 1: Introduction to Machine Learning** Overview of machine learning, pre-processing required for the audio, image and video classification, construction of data loaders for audio, image and video classification, building of deep learning models for audio and video classification, Recent development of deep learning for audio and video classification. **Unit 2: Basics of White-box and Black-Box Adversarial Attacks** Clean sample for deep models, Significance of adversarial sample, crafting of adversarial samples using white box techniques, advantage and disadvantages of white box attack methods, computational (time and memory) analysis of white bo attacks. Construction of adversarial samples using black box attacks, advantage and disadvantage of black box attacks, computational analysis(time, memory) of black box attacks. **Unit 3: Transferability of Adversarial Attacks** Transferability of white box attacks, Transferability of black box attacks, recent trends in enhancing the transferability of adversarial attacks across deep models. **Unit 4: Defense Against Adversarial Attacks** Assessment of robustness of deep models against adversarial attacks, adversarial training of a deep model, Input transformation-based defense techniques against adversarial attacks, encryption-based defense techniques, Failure of adversarial training and challenges. **Unit 5: Advance Attack Techniques:** Poisoning attacks, backdoor attack, backdoor attacks on images, backdoor attack on audio, image and videos. Detection of backdoor attacks. Explaining and understanding the impact of adversarial attacks on deep model using explainability analysis. Explainability assisted adversarial attacks. Explainbility as a defese against adversarial attacks.
4	**Texts/References**	**Textbook(s):** • David J. Miller, Zhen Xiang, Urbana-Champaign, George Kesidis, Adversarial Learning and Secure AI, Cambridge University Press, 2023 • Aneesh Sreevallabh Chivukula, Xinghao Yang, Bo Liu, Wei Liu, • Adversarial Machine Learning: Attack Surfaces, Defense Mechanisms, Learning Theories in Artificial Intelligence, Springer, 2023. **Paper (s):** • Chakraborty, Anirban and Alam, Manaar and Dey, Vishal and Chattopadhyay, Anupam and Mukhopadhyay, Debdeep, A survey on adversarial attacks and defenses, CAAI Transactions on Intelligence Technology, Willey Online, 2021 • Baniecki, Hubert and Biecek, Przemyslaw, Adversarial attacks and defenses in explainable artificial intelligence: A survey, Information Fusion, Elsevier, 2024.

Computer Science Engineering

1	Title of the course (L-T-P-C)	Machine Learning Applications in Wireless Networks (2-1-0-6)
2	Pre-requisite courses(s)	Nil
3	Course content	**Unit 1 Introduction to Wireless Networks and Fundamentals of Machine Learning for Wireless Networks:** Overview of Wireless Networks Cellular, Wi-Fi, and IoT networks; Introduction to Machine Learning: Basic concepts and types of ML algorithms; Challenges and Opportunities in Wireless Networks; Data Preprocessing for Wireless Networks; Supervised, Unsupervised, and Reinforcement Learning in Wireless Networks; Feature Selection and Engineering **Unit 2 Network Performance Optimization and QoS improvement:** Dynamic Resource Allocation in Wireless Networks; Traffic Management and Load Balancing with ML; Predictive Maintenance for Wireless Network Devices; QoS in Wireless Networks; Bandwidth Allocation and Optimization; Prioritizing Traffic with Machine Learning **Unit 3 Network planning, design, and resource management** Capacity Planning with ML; Topology Optimization for Wireless Networks; Case Studies and Practical Applications; ML for Efficient Resource Allocation; Energy Efficiency in Wireless Networks; Optimization of Network Device Placement **Unit 4 Network traffic prediction and traffic engineering:** Traffic Forecasting using ML; Predictive Analytics for Network Traffic; Adaptive routing models; fault prediction and detection for proactive rerouting of traffic; Real-world Applications and Case Studies **Unit 5 Wireless Network Security with Machine Learning:** Intrusion Detection in Wireless Networks; Malware Detection and Prevention; Anomaly Detection using ML; Real-world Applications and Case Studies
4	Texts/References	Machine Learning and Wireless Communications, by Andrea Goldsmith et al., Cambridge University Press, 2022 Machine Learning for Future Wireless Communications, by Fa-Long Luo, Wiley-IEEE press, 2020