Computer Science and Engineering Department
B.Tech in Computer Science and Engineering (2016)

I SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	CE111	Engineering Drawing	1	0	3	5
2	EE101	Electrical Sciences	3	1	0	8
3	HS103	Communicative English for Engineers	2	0.5	1	6
4	MA101	Mathematics – I	3	1	0	8
5	ME110	Workshop-I	0	0	3	3
6	PH103	Physics	3	1	0	8
7	PH110	Physics Laboratory	0	0	3	3
		NCC/ NSS	0	0	0	0
TOTAL			12	3.5	10	41

II SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	CH103	Introduction Chemistry	3	1	0	8
2	CS102	Programming & Data Structure	3	0	0	6
3	CS112	Programming & Data Structure Laboratory	0	0	3	3
4	EE103	Basic Electronics Laboratory	0	0	3	3
5	MA102	Mathematics-II	3	1	0	8
6	ME102	Engineering Mechanics	3	1	0	8
7	CB102 & CE102	Biology & Environmental Studies	3	0	0	6
8	CH110	Chemistry Laboratory	0	0	3	3
TOTAL			15	3	9	45

III SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	MA201	Mathematics – III	3	1	0	8
2	HS2nn	HSS Elective	3	0	0	6
3	CS204	Algorithms	3	0	0	6
4	CS206	Discrete Mathematics	3	1	0	8
5	CS205	Algorithm Laboratory	0	0	3	3
TOTAL			12	2	3	31

IV SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	XX2nn	Open Elective	3	0	0	6
2	HS2nn	HSS Elective	3	0	0	6
3	MA225	Probability Theory & Random Processes	3	1	0	8
4	CS225	Switching Theory	3	0	0	6
5	CS226	Switching Theory Lab	0	0	3	3
6	CS299	Innovative Design Lab	0	0	4	4
TOTAL			12	1	7	33

V SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	XX3nn	Open Elective	3	0	0	6
2	CS303	Formal Language & Automata Theory	3	1	0	8
3	CS321	Computer Architecture	3	0	0	6
4	CS354	Database	3	0	0	6
5	CS355	Database Lab	0	0	3	3
6	CS322	Computer Architecture Lab	0	0	3	3
TOTAL			12	1	6	32

VI SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C

1	**HS3nn**	**HSS Elective**	**3**	**0**	**0**	**6**
2	**CS341**	**Operating System**	**3**	**0**	**0**	**6**
3	**CS358**	**Computer Network**	**3**	**0**	**0**	**6**
4	**CSXXX**	**CS Elective 1**	**3**	**0**	**0**	**6**
5	**CS359**	**Computer Network Lab**	**0**	**0**	**3**	**3**
6	**CS399**	**Seminar**	**0**	**0**	**3**	**3**
7	**CS342**	**Operating Systems Lab**	**0**	**1**	**3**	**5**
TOTAL			**12**	**1**	**9**	**35**

VII SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	XX4nn	Open Elective	3	0	0	6
2	CSXXX	CS Elective -2	3	0	3	9
3	CSXXX	CS Elective -3	3	0	0	6
4	CSXXX	CS Elective -4	3	0	0	6
5	CS491	Project-I	0	0	6	6
TOTAL			12	0	9	33

VIII SEMESTER

Sl.No.	Course No.	Course Title	L	T	P	C
1	CSXXX	CS Elective -5	3	0	0	6
2	CSXXX	CS Elective -6	3	0	0	6
3	CSXXX	CS Elective -7	3	0	0	6
4	CS492	Project-II	0	0	12	12
TOTAL			9	0	12	30

Grand Total	**96**	**11.5**	**65**	**280**

Course No. :CS 102	Name: Programming & Data Structure (PDS)	Credits: 3-0-0-6	Prerequisites: NIL

Syllabus:

Introduction to digital computers; introduction to programming - variables, assignments; expressions; input/output; conditionals and branching; iteration; functions; recursion; arrays; introduction to pointers; structures; introduction to data-procedure encapsulation; dynamic allocation; linked structures; introduction to data structures stacks, queues and trees; time and space requirements.

References:

1. B. W. Kernighan and D. Ritchie, The C Programming Language, Prentice Hall of India (2nd Edition).
2. A. Kelley and I. Pohl, A Book on C, Pearson Education (4th Edition).
3. P.J. Deitel and H.M. Deitel , C How To Program, Pearson Education (7th Edition).

Course No.:CS112	Name: Programming & Data Structure Laboratory (PDS lab)	Credits: 0-0-3-3	Prerequisites: NIL

Syllabus:

Syllabus: Introduction to Unix Commands; Introduction to Program development tools - vi editor, GNU compiler, testing and debugging, etc.; Implementation of programs in C language.

Course No.:CS204	Name: Algorithms	Credits: 3-0-0-6	Prerequisites: CS102 & CS112

Syllabus:

Asymptotic notations, introduction to complexity (time/space) analysis of algorithms. Basic introduction to algorithmic paradigms like divide and conquer, recursion, greedy, dynamic programming, etc. Searching: binary search trees, balanced binary search trees, AVL trees and red-black trees, B-trees, hashing. Priority queues, heaps, Interval trees. Sorting: quick sort, heap sort, merge sort, radix sort, bucket sort, counting sort, etc and their analysis. Graph Algorithms: BFS, DFS, connected components, topological sort, minimum spanning trees, shortest paths, network flow. Reducibility between problems and NP-completeness: discussion of different NP-complete problems.

Texts:

1. M. A. Weiss, Data Structures and Problem Solving Using Java, 2nd Ed, Addison-Wesley, 2002.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, MIT Press, 2001.
3. B. W. Kernighan and D. Ritchie, The C Programming Language, 2nd Ed, Prentice Hall of India, 1988.
4. A. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
5. S. Sahni, Data Structures, Algorithms and Applications in C++, McGraw-Hill, 2001.
6. M. T. Goodrich and R. Tamassia, Algorithm Design: Foundations, Analysis and Internet Examples, John Wiley & Sons, 2001.

Course No.:CS206	**Name:** **Discrete Mathematics**	**Credits: 3-0-0-6**	**Prerequisit es:**

Syllabus:

Propositional logic: Syntax, semantics, valid, satisfiable and unsatisfiable formulas, encoding and examining the validity of some logical arguments; Recurrences, summations, generating functions, asymptotic; Sets, relations and functions: Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction, Finite and infinite sets, countable and uncountable sets, Cantor’s diagonal argument and the power set theorem; Introduction to counting: Basic counting techniques - inclusion and exclusion, pigeon-hole principle, permutation, combination, generating function; Algebraic structures and morphisms: semigroups, groups, subgroups, homomorphisms, rings, integral domains, fields; Introduction to graphs: paths, connectivity, subgraphs, isomorphic and homeomorphic graphs, trees, complete graphs, bipartite graphs, matchings, colourability, planarity, digraphs;

Texts:

1. J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill, 1999.
2. C. L. Liu, Elements of Discrete Mathematics, 2nd Ed, Tata McGraw-Hill, 2000.
3. R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, 2nd Ed, Addison-Wesley, 1994.
4. N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 1974.
5. S. Lipschutz and M. L. Lipson, Schaums Outline of Theory and Problems of Discrete Mathematics, 2nd Ed, Tata McGraw-Hill, 1999.

Lab:

Course No.:CS205	**Name: Algorithm Laboratory**	**Credits: 0-0-3-3**	**Prerequisites: CS102 & CS112**

Syllabus:

The laboratory component will emphasize two areas: Implementation of algorithms covered in class: This will involve running the algorithms under varying input sets and measuring running times, use of different data structures for the same algorithm (wherever applicable) to see its effect on time and space, comparison of different algorithms for the same problem etc. Design of Algorithms: This will involve design and implementation of algorithms for problems not covered in class but related to topics covered in class. The exact set of algorithms to design and implement is to be decided by the instructor. In addition, there will be at least one significantly large design project involving some real world application. An efficient design of the project should require the use of multiple data structures and a combination of different algorithms/techniques. The lab work can be carried out using any programming language.

Course No.:CS225	**Name: Switching Theory**	**Credits: 3-0-0-6**	**Prerequisites: NIL**

Syllabus:

Number Systems, Boolean algebra, logic gates, minimization of completely and incompletely specified switching functions, Karnaugh map and Quine-McCluskey method, multiple output minimization, two-

level and multi-level logic circuit synthesis. Clocks, flip-flops, latches, counters and shift registers, finite-state machine model, synthesis of synchronous sequential circuits, minimization and state assignment, asynchronous sequential circuit synthesis. Programmable logic devices: memory, PLA, PAL. Representation of sequential circuits using ASM charts, synthesis of output and next state functions, data path control path partition-based design.
References: 1. Z. Kohavi, Switching and Finite Automata Theory, 2nd Ed, Tata McGraw-Hill, 1995. 2. M. M. Mano, Digital Design, 3rd Ed, Pearson Education Asia, 2002. 3. S. Brown and Z. Vranesic, Fundamentals of Digital Logic - With Verilog Design, Tata McGraw-Hill, 2002. 4. S. Brown and Z. Vranesic, Fundamentals of Digital Logic - With VHDL Design, Tata McGraw-Hill, 2002 . 5. J. P Uyemura, A First Course in Digital System Design - An Integrated Approach, Vikas Publishing House, 2001.

Lab:

Course No.:CS226	**Name:** **Switching Laboratory**	**Credits: 0-0-3-3**	**Prerequisites:** **NIL**
Syllabus: Combinational logic circuits: Design and implementation of combinational circuits such as ALU and 7-segment LED display driver; Sequential Circuits: Design of sequence generators and detectors, counters, design of ASMs such as, traffic light controllers, lift controllers, etc.			

Lab:

Course No.:CS299	**Name: Innovative Design Lab**	**Credits:** 0-0-3-3	**Prerequisites: NIL**
Syllabus: The objective of this lab would be to encourage and provide support to students for some innovative			

work. The work may focus on inventing a practical solution for a pure Computer Science or multidisciplinary problems. Depending on the nature of the work, it may be carried out in a group or individual mode.

Course No.:CS303	Name: Formal Language & Automata Theory	Credits: 3-1-0-8	Prerequisites: CS206 & CS225

Syllabus:

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of lan- guages. Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata. Context-free languages and pushdown automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. Turing machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumera- tors. Undecidability: Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

References:

1. J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson Education India (3rd edition).
2. K. L. P. Mishra, N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, PHI Learning Pvt. Ltd. (3rd edition).
3. D. I. A. Cohen, Introduction to Computer Theory, John Wiley & Sons, 1997.
4. J. C. Martin, Introduction to Languages and the Theory of Computation, Tata McGraw-Hill (3rd Ed.).
5. H. R. Lewis and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice

Hall, 1997.

6. Garey, D.S., Johnson, G., Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

Course No.:CS321	**Name:** **Computer Architecture**	**Credits: 3-0-0-6**	**Prerequisites: CS225 & CS226**

Syllabus:

Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU - registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study - instruction sets of some common CPUs; Assembly language programming for some processor; Data representation: signed number representation, fixed and floating point representations, character representation. Computer arithmetic - integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division - non-restoring and restoring techniques, floating point arithmetic; CPU control unit design: hardwired and micro-programmed design approaches, Case study - design of a simple hypothetical CPU; Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards; Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs block size, mapping functions, replacement algorithms, write policy; Peripheral devices and their characteristics: Input-output subsystems, I/O transfers - program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes - role of interrupts in process state transitions.

References:

1. David A. Patterson, John L. Hennessy, Computer Organization and Design, Fourth Edition: The Hardware/Software Interface, Morgan Kaufmann; 4 edition, 2011.

2. A. Tenenbaum, Structured Computer Organization, 4th Ed, Prentice-Hall of India, 1999.

3. W. Stallings, Computer Organization and Architecture: Designing for Performance, 6th Ed, Prentice Hall, 2005.

4. J. Hennessy and D. Patterson, Computer Architecture A Quantitative Approach, 3rd Ed, Morgan Kaufmann, 2002.

Lab:

Course No.: CS322	Name: Computer Architecture Lab	Credits: 0-0-3-3	Prerequisites: CS225 & CS226

Syllabus:

Familiarization with assembly language programming; Synthesis/design of simple data paths and controllers, processor design using HDL like verilog/vhdl; Interfacing - DAC, ADC, keyboard-display modules, etc. Development kits as well as Microprocessors/PCs may be used for the laboratory, along with design/simulation tools as and when necessary.

Course No.:CS354	Name: Database	Credits: 3-0-0-6	Prerequisites: CS206

Syllabus:

Database system architecture: Data Abstraction, Data Independence, Data Definition and Data Manipulation Languages; Data models: Entity-relationship, network, relational and object oriented data models, integrity constraints and data manipulation operations; Relational query languages: Relational algebra, tuple and domain relational calculus, SQL and QBE; Relational database design: Domain and data dependency, Armstrongs axioms, normal forms, dependency preservation, lossless design; Query processing and optimization: Evaluation of relational algebra expressions, query equivalence, join strategies, query optimization algorithms; Storage strategies: Indices, B-trees, hashing; Transaction processing: Recovery and concurrency control, locking and timestamp based schedulers, multiversion and optimistic Concurrency Control schemes; Recent Trends: XML Data, XML Schema, JSON and "NoSQL Systems, etc,.

References:

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan, Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai, An Introduction to Database Systems, Galgotia.
4. J. D. Ullman, Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe, Fundamentals of Database Systems, Addison-Wesley.
6. Serge Abiteboul, Richard Hull and Victor Vianu, Foundations of Databases. Addison-Wesley

Lab:

Course No.: CS355	Name: Database Laboratory	Credits: 0-0-3-3	Prerequisites: NIL
Syllabus: Database schema design, database creation, SQL programming and report generation using a commercial RDBMS like ORACLE/SYBASE/DB2/SQL-Server/INFORMIX. Students are to be exposed to front end development tools, ODBC and CORBA calls from application Programs, internet based access to databases and database administration.			

Course No.:CS341	Name: Operating System Theory	Credits: 3-0-0-6	Prerequisites: CS102 & CS321
Syllabus: Process Management: process; thread; scheduling. Concurrency: mutual exclusion; synchronization; semaphores; monitors; Deadlocks: characterization; prevention; avoidance; detection. Memory Management: allocation; hardware sup- port; paging; segmentation. Virtual Memory: demand paging; replacement; allocation; thrashing. File Systems and Implementation. Secondary Storage: disk structure; disk scheduling; disk management. (Linux will be used as a running example, while examples will drawn also from Windows NT/7/8.); Advanced Topics: Distributed Systems. Security. Real-Time Systems.			
References: 1. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, 9th Ed, John Wiley & Sons, 2010. 2. A. S. Tenenbaum, Modern Operating Systems, 2nd Ed, Prentice Hall of India, 2001. 3. H. M. Deitel, P. J. Deitel and D. R. Choffness, Operating Systems, 3rd Ed, Prentice Hall, 2004. 4. W. Stallings, Operating Systems: Internal and Design Principles, 5th Ed, Prentice Hall, 2005. 5. M. J. Bach, The Design of the UNIX Operating System, Prentice Hall of India, 1994. 6. M. K. McKusick et al, The Design and Implementation of the 4.4 BSD Operating System, Addison Wesley, 1996.			

Lab:

Course No.: CS342	Name: Operating System Laboratory	Credits: 0-1-3-5	Prerequisites: CS102 & CS112

Syllabus:

Programming assignments to build different parts of an OS kernel.

Course No.:CS358	Name: Computer Network	Credits: 3-0-0-6	Prerequisites: nil

Syllabus:

Evolution of computer networks; Physical Layer: Theoretical basis for data communication, transmission media and impairments, switching systems Medium Access Control Sublayer: Channel allocation Problem, multiple access protocols, Ethernet Data link layer: Framing, HDLC, PPP, sliding window protocols, error detection and correction Network Layer: Internet addressing, IP, ARP, ICMP, CIDR, routing algorithms (RIP, OSPF, BGP); Transport Layer: UDP, TCP, flow control, congestion control; Introduction to quality of service; Application Layer: DNS, Web, email, authentication, encryption.

References:

1. Peterson & Davie, Computer Networks, A Systems Approach: 5th Edition
2. William Stallings Data and Computer Communication, Prentice Hall of India.
3. Behrouz A. Forouzan, Data Communication and Networking, McGraw-Hill.
4. Andrew S. Tanenbaum, Computer Networks, Prentice Hall.
5. Douglas Comer, Internetworking with TCP/IP, Volume 1, Prentice Hall of India.
6. W. Richard Stevens, TCP/IP Illustrated, Volume 1, Addison-Wesley.

Lab:

Course No.: CS359	Name: Computer Network Lab	Credits: 0-0-3-3	Prerequisites: CS101 & CS110

Syllabus:

Simulation experiments for protocol performance, configuring, testing and measuring network devices and parameters/policies; network management experiments; Exercises in network programming.

Course No.: CS491	Name: Project-I	Credits: 0-0-6-6	Prerequisites: NIL

Syllabus:

The project can span the course Project-II. Hence it is expected that the problem specification and the milestones to be achieved in solving the problem are clearly specified.

Course No.: CS492	Name: Project-II	Credits: 0-0-12-12	Prerequisites: NIL

Syllabus:

The students who work on a project are expected to work towards the goals and milestones set in course Project-I. At the end there would be demonstration of the solution and possible future work on the same problem. A dissertation outlining the entire problem, including a literature survey and the various results obtained along with their solutions is expected to be produced.

Elective

Course No.:CS547	Name: Foundation of Computer Security	Credits: 3-0-0-6	Prerequisites: nil

Syllabus:

Introduction to Computer Security and Privacy : security and privacy; types of threats and attacks; methods of defense Program Security: Secure programs; nonmalicious program errors; malicious code; controls against program threats Operating System Security: Methods of protection; access control; user authentication Network Security: Network threats; firewalls, intrusion detection systems Internet Application Security and Privacy: Basics of cryptography; security and privacy for Internet applications (email, instant messaging, web browsing); privacy-enhancing technologies Database Security and Privacy: Security and privacy requirements; reliability, integrity, and privacy; inference;

Note: Familiarity with CS 341 Operating Systems and CS 101 Programming in C, is desirable

References:

1. Security in Computing, 4thedition. Charles P. Pfleeger and Shari Lawrence Pfleeger Prentice-Hall, 2007. Or later
2. Introduction to Computer Security Matt Bishop, Addison-Wesley 2005
3. Published papers in this area will be discussed and uploaded in the course-web

Elective

Course No.:CS503	Name: Advances in Algorithms	Credits: 3-0-0-6	Prerequisites: CS101, CS204, CS206

Syllabus:

Algorithmic paradigms: Dynamic Programming, Greedy, Branch-and-bound; Asymptotic complexity, Amortized analysis; Graph Algorithms: Shortest paths, Flow networks; NP-completeness; Approximation algorithms; Randomized algorithms; Online algorithms; Streaming algorithms; Linear programming;

Special topics: Geometric algorithms (range searching, convex hulls, segment intersections, closest pairs), Numerical algorithms (integer, matrix and polynomial multiplication, FFT, extended Euclid's algorithm, modular exponentiation, primality testing, cryptographic computations), Internet algorithms (text pattern matching, tries, information retrieval, data compression, Web caching).

References:

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, 2nd edition, Prentice-hall Of India Pvt.. Ltd, (2007)

2. J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley, (2008)
3. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press, (1995)
4. Vijay Vazirani, Approximation Algorithms, Springer, (2004)
5. Soumen Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data, Elsevier India Private Limited, (2005)
6. Technical papers from major reputed journals in the area of algorithms design

Elective

Course No.:CS505	Name: Advanced Graph Theory	Credits: 3-0-0-6	Prerequisites: nil

Syllabus:

Basic Concepts: Graphs and digraphs, incidence and adjacency matrices, isomorphism, the automorphism group;

Trees: Equivalent definitions of trees and forests, Cayleys formula, the Matrix-Tree theorem, minimum spanning trees;

Connectivity: Cut vertices, cut edges, bonds, the cycle space and the bond space, blocks, Menger s theorem;

Paths and Cycles: Euler tours, Hamilton paths and cycles, theorems of Dirac, Ore, Bondy and Chvatal, girth, circumference, the Chinese Postman Problem, the Travelling Salesman problem, diameter and maximum degree, shortest paths;

Matchings: Berges Theorem, perfect matchings, Halls theorem, Tuttes theorem, Konigs theorem, Petersens theorem, algorithms for matching and weighted matching (in both bipartitie and general graphs), factors of graphs (decompositions of the complete graph), Tuttes f-factor theorem;

Extremal problems: Independent sets and covering numbers, Turans theorem, Ramsey theorems; Colorings: Brooks theorem, the greedy algorithm, the Welsh-Powell bound, critical graphs, chromatic polynomials, girth and chromatic number, Vizings theorem; Graphs on surfaces: Planar graphs, duality, Eulers formula, Kuratowskis theorem, toroidal graphs, 2-cell embeddings, graphs on other surfaces;

Directed graphs: Tournaments, directed paths and cycles, connectivity and strongly connected digraphs, branchings;

Networks and flows: Flow cuts, Max flow min cut theorems, perfect square;

Selected topics: Dominating sets, the reconstruction problem, intersection graphs, perfect graphs, random graphs.

References:

1. D.B.West: Introduction to Graph Theory, Prentice-Hall of India/Pearson, 2009
2. J.A.Bondy and U.S.R.Murty: Graph Theory, Springer, 2008.

3. R.Diestel: Graph Theory, Springer(low price edition) 2000.
4. F.Harary: Graph Theory, Narosa, (1988)
5. C. Berge: Graphs and Hypergraphs, North Holland/Elsevier, (1973)

List of Other Approved Electives:

- CS561: Artificial Intelligence
- CS544: Introduction to Network Science
- CS542: Software Testing
- CS543: Distributed Systems
- CS528: CAD for VLSI
- CS548: Wireless Networks
- CS549: Computer and Network Security
- CS508: Formal methods for analysis and verification
- CS743: Advanced topics on Database
- CS502: Pattern Recognition
- CS563: Natural Language Processing
- CS564: Foundations of Machine Learning