School of Computer & Information Sciences

M.Tech (Computer Science)

SCHOOL OF COMPUTER & INFORMATION SCIENCES

Vision Statement:

- To invent, create and bring computing technology solutions to the common man, to the privileged and underprivileged sections of India, to bridge the digital divide and eradication of computer ignorance and digital illiteracy and to build a prosperous and technologically advanced nation.

Mission Statements:

MS-1: To pursue academic and research excellence, nationally and internationally

MS-2: To provide training, advisory, and consultancy to all the stakeholders.

MS-3: To lead the efforts in creative and newer modes of instruction delivery & supervision

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (Computer Science)

Program Educational Objectives (PEOs)

PEO-1: Produce Post graduates who can contribute to the Research & Development effectively

PEO-2: To provide students a deep insight into cutting edge technologies and tools.

PEO-3: To create globally competent technocrat's with exposure to Scientific & Engineering aspects of development

PEO-4: To work collaboratively on multi-disciplinary projects and exhibit high levels of professional & ethical values

PEO-5: Create awareness of societal problems and its impact

Mapping Program Educational Objectives (PEOs) with Mission Statements (MS)

	MS-1	MS-2	MS-3
PEO-1	3	2	1
PEO-2	2	3	1
PEO-3	3	2	1
PEO-4	2	1	3
PEO-5	1	2	3

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences

Name of the Academic Program: M.Tech(Computer Science)

Program Outcomes (POs)

PO-1: To independently carry out research/investigation and development work to solve practical problems
PO-2: To be able to write and present a substantial technical report/document
PO-3: To demonstrate knowledge and understanding of engineering principles and apply the same in solving the problems faced by society.
PO-4: To create, select, learn and apply appropriate techniques, resources, and advanced tools, including modeling and prediction with an understanding of limitations
PO-5: To recognize the opportunities and contribute to collaborative-multidisciplinary scientific research to achieve common goals.
PO-6: To acquire professional and intellectual integrity, professional ethics code of conduct and understand the responsibility to contribute to the society for sustainable development

Program Specific Outcomes (PSOs)

PSO-1: To analyse, design and assess technical challenges in advanced architectures and networks and develop solutions to optimize the resources and increasing the performance of the systems.
PSO-2: To design, formulate and solve architectural features of threaded algorithms, GPU's etc , extract maximum performance in multicore, shared memory execution and deploy large scale parallel algorithms on tightly coupled parallel systems using message passing paradigms.
PSO-3: To analyse and assess various functional and technical security challenges in protecting various digital assets and infrastructure in the internet era and to design and develop innovative technological solutions for the same

Mapping of Program Outcomes (POs) and Program Specific Outcomes (PSOs) with Program Educational Objectives (PEOs)

	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
PO-1	3		1	2	
PO-2	2	1		3	
PO-3	1	3		2	
PO-4	2		1	3	
PO-5			1	3	2
PO-6		1	2		3
PSO-1	3			2	1
PSO-2	1	3			2
PSO-3	1		3		2

Mapping of Program Specific Outcomes (PSOs) where applicable.

Write '3' in the box for 'high-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low-level' mapping.

School of Computer & Information Sciences
M.Tech (Computer Science)
Scheme

I-Semester					
Core			Optional Core/Elective		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CS401	Advanced Operating Systems	4	CS421	Introduction to Data Compression	3
CS402	Algorithms	4	CS422	Distributed Computing	3
	Optional core-I	¾	CS423	Human Computer Interaction	3
	Optional core-II	¾	CS424	Parallel Computing	4
	Elective -I	¾	CS425	Cryptography	3
CS403	*IT Lab –I*	2	CS426	Block chain Technologies	3
CS404	*DS & Programming Lab*	2	CS427	Advanced Computer Architecture	3
			CS428	Graph Theory	3
		21/24			
III-Semester					
Code	**Course Title**	**Credits**			
	Project	6			

School of Computer and Information Sciences

Name of the Academic Program: M.Tech(Computer Science) (M.Tech-1)

Course Code: CS 401 Title of the Course: Advanced Operating Systems
L-T-P : 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): -- Basic OS course

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Discuss the ways system calls work.(Understand)
- CO2: Develop basic process management tasks such as scheduling, deadlock avoidance algorithms. (Create)
- CO3: Develop paging algorithm.(Create)
- CO4: Construct simple device drivers. (Create)
- CO5: Describe different file systems in existence and learn the pros and cons of the various systems. (Understand)
- CO6: Examine real world OS scheduling algorithms such as those used in Linux and Windows. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1				3			2		
CO2			3	2			1		
CO3	1			3	2		1		
CO4			2	3				1	
CO5		2		3			1		
CO6	2					3		1	

Detailed Syllabus:

UNIT - I: Introduction and Operating System Structures

Operating Systems Functionality, Computer Organization and Architecture, OS Operations, Kernel Data Structures, OS Services, User interfaces to OS, Programmer interfaces to OS, OS Structure, System Boot.

UNIT - II: Process and Thread Management

Process Concept, Process operations, Process Scheduling, Extended Process State Diagram, Process Context Switch in detail; Inter process Communication: Pipes, Named Pipes, Shared Memory; Process Synchronization: Signals, Mutexes, Semaphores, Monitors; Thread Management: thread creation, thread scheduling, thread synchronization; Deadlocks: Resource Allocation Graphs, deadlock detection, prevention and avoidance, recovery from deadlock.

UNIT - III: Memory Management

Memory allocation techniques: paging and segmentation, Swapping, structure of the page table; Virtual memory: demand paging, copy-on-write, Page replacement, allocation of frames, kernel memory allocation, thrashing, memory-mapped files, Translation-Lookaside Buffer (TLB).

UNIT - IV: File System Management

Disk management: formatting, boot block, swap-space management, RAID structure; Disk scheduling algorithms: elevator, C-SCAN; File concept, Access methods, Directory structure, file sharing, protection, file system structure; file system implementation: file system metadata storage structures such as inode, allocation methods, free space management, efficiency and performance including disk cache and recovery from failures.

UNIT - V: I/O Management

I/O devices: polling, interrupt-driven, DMA; Application I/O interface: character and block devices, network devices; clocks and timers, nonblocking and asynchronous I/O, vectored I/O; Kernel I/O interface: I/O scheduling, Buffering, Caching.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne. Operating System Concepts, 9th edition, Wiley.
2. Charles Crowley. Operating Systems: A Design-Oriented Approach, Prentice-Hall India.
3. W. Richard Stevens, . Advanced Programming in Unix Environment, Pearson Education.
4. W. Richard Stevens. Unix Network Programming, vol. 2, Pearson Education.
5. William Stallings. Operating Systems: Internals and Design Principles, Pearson Education.
6. Maurice J. Bach. The Design of the Unix Operating System, Prentice-Hall India.
7. Robert Love. Linux Kernel Development, Pearson Education.
8. Thomas Anderson and Michael Dahlin. Operating Systems: Principles and Practice, 2nd edition, Recursive Books.

School of Computer and Information Sciences

Name of the Academic Program: M.Tech((Computer Science) (M.Tech-I)

Course Code: CS402 Title of the Course: Algorithms
L-T-P: 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): Data Structures in under graduate level, discrete mathematical structures, knowledge of sorting algorithms and basic search strategies

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO-1: Assess the inherent structure/hardness of a problem (Evaluate)
- CO-2: Select an appropriate strategy to solve a problem (Understand)
- CO-3 Design an algorithm that suits the time complexity requirements of the problem. (Create)
- CO-4: Estimate the time and space complexities of an algorithm along with the necessary mathematical proofs when necessary. (Evaluate)
- CO-5: Devise algorithms by choosing appropriate data structures (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1	2			3			1		
CO2				3		1		2	
CO3			3				2		
CO4		3			2		1		
CO5				3		2			

Detailed Syllabus

UNIT-I: Analysis of Algorithms: Asymptotic Notation; Best, worst and average case analysis of algorithms; Solving recurrence relations using substitution method, generating functions, Master's theorem etc. Warm-up to complexity analysis: Heap data structure, priority queue application, Best, worst and average case analysis of a few sorting algorithms like heap sort, insertion, bubble, selection, counting and radix sort algorithms. Strategies for problem solving

UNIT-II: Divide and Conquer strategy: Time complexity analysis for Merge Sort and Quick Sort Algorithms

UNIT-III: Greedy strategy: Theoretical foundation of greedy strategy: Matroids Algorithms for solving problems like Knapsack Problem (Fractional), Minimum Spanning Tree problem; Shortest Paths, Job Scheduling, Huffman's code etc along with proofs of corrections and complexity analysis

UNIT-IV: Dynamic Programming strategy: Identify situations in which greedy and divide and conquer strategies may not work. Understanding of optimality principle. Technique of memorization. Applications to problems like Coin change, 0/1 and 0/n- Knapsack, Shortest Paths, Optimal Binary Search Tree (OBST), Chained Matrix Multiplication, Traveling Salesperson Problem (TSP) etc.

UNIT-V: Backtracking and Branch & Bound strategies: State space tree construction, traversal techniques and solving problems like 0/1 and 0/n knapsack, TSP, Applications of Depth First Search: Topological sorting, Finding strongly connected components and game problems.

UNIT-VI: Theory of NP-Completeness: Complexity classes of P, NP, NP-Hard, NP-Complete, Polynomial reductions, Cook's theorem. Discussion of problems: Satisfiability(SAT), CNF-SAT, Min-Vertex Cover, Max-Clique, Graph Coloring, NP-Completeness proofs.

Reference Books:

1. Introduction to Algorithms-T.Cormen, C.E.Leiserson, R.L.Rivest, PHI, 3rdEdition 2009.
2. Algorithms- R.Johnsonbaugh and M.Schaefer, Pearson, 2004.
3. Fundamentals of Algorithmics - G.Brassard and P.Bratley, PH, 1996
4. The Algorithm Design Manual- Steven S. Skiena, Springer, 2009

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (Computer Science) (MTECH-I)

Course Code: CS404 Title of the Course: Data Structures & Programming Lab

L-T-P: 0-0-3 Credits : 2

Prerequisite Course / Knowledge (If any): Programming and Data Structures at under graduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Solve a problem by choosing appropriate data structures in C programming language (Apply)
- CO-2: Select suitable data structure for an idea and propose solution using C Programming Language (Analyze)
- CO-3: Analyze the time taken to solve the problem by using C programming language (Analyze)
- CO-4: Assess the solution in terms of efficiency, modularity and well-documented programs in C under Linux environment (Evaluate)

Mapping of Course Outcomes (COs) with Program Outcomes (POs)and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1	3		1						
CO2			1	3					
CO3		2							
CO4			2		1		3		

Detailed Syllabus:

UNIT-I: Implementing Stacks and types of Queues as dynamic data structures using linked lists and their applications

UNIT-II: Binary Tree, Binary search trees & Traversals of BST, balanced trees - AVL Trees and their applications

UNIT-III: Sorting Techniques, Basic Searching Techniques, Hashing-Collision Resolution and closed hashing.

UNIT-IV: Graphs: Representations (Matrix and Adjacency List), basic traversal techniques: Depth First Search , Breadth First Search, Implementation of Kruskal Algorithm, Dijkstra Algorithm, Spanning and Minimal Spanning Trees.

UNIT-V: Multi link Structures, B Trees and B+ Trees and their applications.

Reference Books:

1. Horowitz, E., and Sahni.S: Fundamentals of Data structures. Computer Science Press, 1978.
2. Tanenbaum, A.M., and Augenstein, M.J.: Data Structures with Pascal, Prentice - Hall International, 1985.
3. Stubbas, D.: Data Structures with Abstract Data Types and Modula2, Brooks & Cole Pub. Co. 1987.
4. Trembley & Sorenson: An Introduction to Data Structures with Applications; Tata McGraw Hill
5. Kruse, R.L., Leung, B.P., and tondo, C.L.: Data Structures and Program Design in C; Prentice-Hall of India 1999.

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (I-Sem)

Course Code: CS421 Title of the Course: Introduction to Data Compression

L-T-P: 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Students are expected to have completed Computer Algorithms, must have done Basic Calculus course, and has basic understanding of Probability

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Test whether a code is uniquely decodable or not (Analyze)
- CO-2: Compare codes using measures of performance (Evaluate)
- CO-3: Design Loss Less compression codes such as Huffman code, Non-binary Huffman code, Extended Huffman code, Adaptive Huffman code, Golomb code, Tunstall code, Arithmetic code, Lempel Ziv (LZ77, LZ78, LZW) codes, etc. (Create)
- CO-4: Explain the advantages of an Arithmetic code over an extended Huffman code. (Understand)
- CO-5: Evaluate the entropy of a source given its probability model, determine whether a given code is optimum code or not, determine bounds on the average length of an optimum code. (Evaluate)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1				3	1				2
CO2			3			2			1
CO3	2					1			3
CO4			3	2					1
CO5		1				2			3

Detailed Syllabus:

UNIT 1: Introduction to Data Compression, Mathematical Preliminaries for Lossless Compression, Huffman Coding

Lossless and Lossy Compression, Self information of an event, Entropy of a source, Test for Unique Decodability, Uniquely Decodable Codes, Prefix Codes, Kraft MacMillan Inequality

UNIT 2: Huffman Coding

Design of Huffman Codes, Optimality of Huffman Codes, Average Length of Huffman Codes, Extended Huffman Codes, Non-binary Huffman Codes, Adaptive Huffman Coding, Golomb Codes, Tunstall Codes

UNIT 3: Arithmetic Coding

Motivation for Arithmetic Coding, Coding a Sequence, Generating a Tag, Deciphering the Tag, Generating a Binary Code, Uniqueness and Efficiency of the Arithmetic Code, Algorithm Implementation Issues, Issues related to Integer Implementation, Comparison of Huffman and Arithmetic coding.

UNIT 4: Dictionary Techniques

Dictionary Techniques: Static Dictionary Techniques, Adaptive Dictionary Techniques, Universal Compression, LZ, LZW, etc. algorithms

UNIT 5: Context Based Compression

Introduction, Prediction with partial match (ppm) algorithm and its variations, the Burrows-Wheeler Transform, Move-to-Front Coding

Reference Books:

1. Khalid Sayood (2018), *Introduction to Data Compression*, 5th Edition, Morgan Kaufmann Publishers, Singapore.
2. David Salomon (2004), Data Compression: The Complete Reference, 3rd Edition, Springer, New York.

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (I-Sem)

Course Code: CS422 Title of the Course: Distributed Computing

L-T-P: 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that students are aware of basics of Computer Networks especially TCP/IP Protocol Suite and are comfortable in Java Programming

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO-1: Explain the different Distributed Computing paradigms along with applications (Understand)
- CO-2: Design and Develop TCP Client/Server based Distributed Applications using Java Socket API (Apply)
- CO-3: Design and Develop UDP Unicast and Multicast Client/Server based Distributed Applications using Java Socket API (Apply)
- CO-4: Assess the importance of Abstraction in Distributed Object based Applications using Java RMI (Understand)
- CO-5: Explain the differences among Client-Server based Distributed Systems and Peer-to-Peer based Distributed Systems and designing aspects of Peer-to-peer Distributed Systems (Understand)
- CO-6: Ability to design Distributed Algorithms and assess the roles of Synchronization, Communication and Coordination in the development of Distributed Algorithms (Evaluate Level)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1				1			2	3	
CO2			3				1	2	
CO3			3				1	2	
CO4				1			3	2	
CO5					1		2	3	
CO6	2					1		3	

Detailed Syllabus:

UNIT-I:

Introduction- Distributed computing Concepts, Basic network concepts, Basic operating system concepts.
Inter process Communication- Basic model, Primitives (operations): connect, send, receive, disconnect, Connection -oriented/connectionless, Data marshalling: data flattening, data representation, serialization, Event synchronization, Event diagram, sequence diagram.
Distributed Computing Paradigms- Message passing, client server, P2P, message system, RPC, Distributed Object Paradigm, Object Space, Mobile agent, Network services, Groupware paradigms, SOA, Overview & comparison of each paradigm.

Unit-II

The Socket API- The basic model, Stream-mode (connection-oriented) socket, Datagram socket (connectionless) socket, Java socket API, Using socket to implement a client. Using sockets to implement a server, A simple middleware using sockets, Secure sockets and the Java secure socket extension API.
The Client-server Paradigm- The daytime protocol and a sample client -server suite, The echo protocol and a sample client-server suite, Connection-oriented client-server, Connectionless client-server, Iterative server and concurrent server, Stateful server and stateless server.
Group Communications- Unicast versus multicast, Basic model of group communications, The Java multicast API. Sample multicast sender program, Sample multicast listener program, Multicast and message ordering, Reliable multicast/broadcast.

Unit-III

Distributed objects- Message passing versus distributed objects, The basic model, Remote procedure call, Remote method invocation, RMI Architecture, RMI Object Oriented Semantics, Dynamic Downloading of Classes, RMI Security Manager.
Peer-to-Peer Systems- Introduction, Peer-to-Peer Middleware, Routing Overlays, Case Study: Chord.
Case Studies in Distributed Algorithms- Distributed Consensus (Paxos), Ordered Multicast Algorithms, Replication (Gossip), Google MapReduce.

Reference Books:

1. M. L. Liu (2004), ***Distributed Computing: Principles and Applications***, First Edition, Pearson Publications.
2. George Coulouris, Jean Dollimore , Tim Kindberg and Gordon Blair (2017), ***Distributed Systems: Concepts and Design***, Fifth Edition, Pearson Publisher.
3. Sukumar Ghosh (2014), ***Distributed Systems: An Algorithmic Approach,*** Second Edition, Chapman and Hall/CRC Press.
4. Research Publications of Distributed Algorithms

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (I-Sem)

Course Code: CS424 Title of the Course: Parallel Computing

L-T-P: 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): Knowledge of Introductory Algorithms, Networks, Java/C/C++, and Unix/Linux.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO1: Illustrate in parallel for the solution of a given problem. Should be able to understand parallel algorithm paradigms (Apply level)
- CO2: Analyze parallel time and processor's complexity with different models of Parallel Algorithms and design efficient parallel algorithms for given problems (Analyze level)
- CO3: Apply the Message Passing Interface/Compiler directives for getting solution (Apply level)
- CO4: Describe real implementation issues and write algorithms for the target machine (Multiprocessors/Multicomputer) (understand level)
- CO5: Explain the use of certain parallel design paradigm, architecture using different parameters for a given problem solution (Evaluate level)
- CO6: Develop skill to learn how to write parallel library (Create level)
- CO7: Develop skill to learn how to write parallel modules on GPU's (Create level)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			1	2				3	
CO2			2	1				3	
CO3			1				2	3	
CO4				1			3	2	
CO5			1	2				3	
CO6	2	1						3	
CO7	2	1						3	

Detailed Syllabus:

UNIT-I: Introduction to Parallel Computing: Why Parallel Computing & Scope of Parallel Computing, Sieve of Eratosthenes, Control and Data Approach, PRAM model of parallel computation, Design paradigms of Parallel Computing, examples, Bulk Synchronous Parallel (BSP) model, algorithms on PRAM and BSP model.

UNIT-II: Practical Parallel Programming Paradigms**:** Foster's design paradigm for Multi computing programming, Programmability Issues, Programming Models: Message passing, Message passing standards: PVM (Parallel Virtual Machine), MPI (Message Passing Interface) and its routines, Advanced Features of MPI, Load balancing techniques. Programming on Multiprocessors: Introduction to OpenMP (History, Overview, Programming Model, OpenMP Constructs, Performance Issues and examples, Explicit Parallelism: Advanced Features of OpenMP).

UNIT-III: Threading on Intel Multi-Core Processors: Hardware-based Threading, Hyper-Threading Technology , Difference between Multiprocessor and Hyper-Threading, Technology, Hyper-Threading Technology Architecture, Multi-Core Processors, Architectural Details, Comparison between Multiprocessors and Multi-Core, Processors, Multiple Processor Interaction, Inter-Processor Communication and Multi-threaded Programming, Power Consumption, Power Metrics.

UNIT-IV: Introduction to Heterogeneous Multi-Core Processors: Introduction to Many cores Programming, Cell Processor Multinode Computing, The Early Days of GPGPU Coding, GPU Hardware, Alternatives to CUDA, OpenCL, Direct ComputeCPU alternatives, Directives and libraries, Understanding Parallelism with GPUs.

UNIT-V: Heterogeneous Multi-Core Programming with CUDA: Introduction to GPU Computing, CUDA Programming Model, CUDA API, Simple Matrix Multiplication in CUDA, CUDA Memory Model, Shared Memory Matrix Multiplication, Additional CUDA API Features, Threading Hardware, Memory Hardware, Memory Bank Conflicts, Parallel Thread Execution, Control Flow, Precision.

Suggested Readings::

1. Quinn, M. J. (2003), *Parallel Programming in C with MPI and OpenMP*, First ed., McGraw-Hill Education.
2. Bary Wilkinson and Michael Allen (1999),*Parallel Programming Techniques using networked of workstations and Parallel Computers*, First edition,Pearson.
3. W. Gropp, E. Lusk, N. Doss, A. Skjellum(1996), *A high performance portable implementation of the message passing Interface (MPI) standard*, Parallel Computing 22 (6).
4. Gibbons, A., W. Rytter (1989), *Efficient Parallel Algorithms*, first Edition Cambridge Uni. Press.
5. AnanthGrama et al. (2004), An Introduction to Parallel Computing: Design and Analysis of Algorithms, Second edition, Pearson Education India.
6. David B. Kirk, Wen-mei W. Hwu(2010), *Programming Massively Parallel Processors: A Hands-on Approach*, First Edition, Morgan Kaufmann (This book is only on NVIDIA GPUs and CUDA programming despite its title)
7. Jason Sanders and Edwards Kandrot(2011), *CUDA by Example: An Introduction to General-Purpose GPU Programming,* First Edition, Addison-Wesley.
8. Shane Cook, Morgan Kaufmann (2012), *CUDA Programming A Developer's Guide to Parallel Computing with GPUs,* First Edition, Morgan Kaufmann.

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (I-Sem)

Course Code: CS 425 Title of the Course: Cryptography

L-T-P: 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): None

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO1: Describe various mathematical techniques used in cryptology (Understand)
- CO2: Illustrate the design techniques of various cryptographic algorithms through the analysis of classic ciphers. (Apply)
- CO3: Apply different mathematical techniques in various symmetric and asymmetric cryptographic algorithms (Apply)
- CO4: Evaluate the need for digital signatures and their importance in modern electronic era. (Evaluate)
- CO5: Examine how a mathematical hard problem can be used to construct into a public key cryptosystem. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			2	1					3
CO2				2	1				3
CO3	2		1						3
CO4	1				2				3
CO5	2					1			3

Each Course Outcome (CO) may be mapped with one or more Program Outcomes (POs). Write '3' in the box for 'High-level' mapping, 2 for 'Medium-level' mapping, 1 for 'Low'-level' mapping

Mapping with PSOs, where applicable.

Detailed Syllabus:

UNIT-I: OVERVIEW, HISTORY AND CLASSICAL CIPHERS

Cryptography, steganography and cryptanalysis; History and development of cryptography; Classical cryptosystems: shift, substitution and Vigen´ere ciphers; Attacks on shift, substitution and Vigenere ciphers; Enigma cryptosystem and Role of WW-II; Designing a provably secure system, One -Time pads.

UNIT-II: SYMMETRIC KEYCRYPTOSYSTEMS AND GSM SECURITY

Basics of number theory and algebra; Introduction to information theory, Shannon's axioms; DES and AES; Encryption in GSM communications, A5 family of algorithms.

UNIT-III: ASYMMETRICKEY CRYPTOSYSTEMS AND DIGITALSIGNATURES

Prime numbers, factorization and discrete algorithms; RSA and El Gamal cryptosystems; Signature schemes, hash functions and secret sharing schemes.

UNIT-IV: INTRODUCTIONTOCRYPTANALYSIS

Known plaintext, known cipher text, chosen plaintext and chosen cipher text attacks, man-in-the-middle attacks; Attacks on DES and AES, differential cryptanalysis; Attacks on RSA; Attacks on El Gamal; Attacks on A5 family.

UNIT-V: ADVANCED TOPICS

Zero knowledge proofs; Pseudo-random number generators; Industry standards and practices.

Reference Books

1. Douglas Stinson. Cryptography: Theory and Practice, Third Edition or higher, Chap-man &Hall/CRC (Indian Edition) 2011.
2. Alfred Menezes, Paul C. van Oorschot and Scott A. Vanstone. Handbook of Applied Cryptography, CRC Press (2001). Free download in PDF available from http://cacr.uwaterloo.ca/hac/

Additional Reading

1. Johannes Buchmann. Introduction to Cryptography, Springer Pubs., 2nd Edition (2004) .

2. Lawrence C. Washington. Elliptic Curves, Number Theory and Cryptography, Chapman & Hall/CRC 2nd Edition (2008). 3. Simon Singh. The Code Book, 4th Estate Pubs. (2002)

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (M.Tech-I)

Course Code: CS 426 Title of the Course: Blockchain Technologies

L-T-P: 3-0-0 Credits: 3

Prerequisite Course / Knowledge (If any): Nil

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Describe various cryptographic techniques used in Block chains (Understand)
- CO2: Discuss the architecture and functionality of how Public block chains work (Understand)
- CO3: Develop code for simulation of Public Block chains (Create)
- CO4 : Describe the architecture and functionality of how Private block chains work (Understand)
- CO5: Develop code for simulation of Private Block chains (Create)
- CO6: Examine applications of block chains in various real time domains (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1		1	3						2
CO2				3	2				2
CO3				2		1			3
CO4		1	3						2
CO5	2	1							3
CO6	2					1			3

Detailed Syllabus::

UNIT – I - Preliminaries:

Hash Functions, Properties of hash functions, Security of hash functions, SHA-160, SHA-256, Ethash, Scrypt etc. Digital Signatures, RSA-based signature scheme, ECDSA, multi-signatures, achieving consensus in distributed systems.

UNIT – II - Permissionless (Public) Blockchain:

How bitcoin works, Public key cryptography and crytocurrency, bitcoin transactions in detail, bitcoin script, the bitcoin network, Simulation of Permissionless Blockchains.

UNIT – III - Permissionless Blockchain continued:

The bitcoin blockchain architecture in detail, characteristics of blockchain, bitcoin mining, consensus in bitcoin, Smart contract, comparing bitcoin blockchain with other permissionless blockchains like Ethereum, Ripple, Litecoin etc.

UNIT – IV - Permissioned (Private) Blockchains:

Introduction to Hyperledger fabric, characteristics of fabric, fabric architecture in detail, fabric components (chaincode, MSP, peers, endorsing peers, ordering service, committing peers, state DB), Endorsement policies, achieving consensus in fabric, channels, Comparing hyperledger fabric with other permissioned blockchains, Simulation of Permissioned Blockchains.

UNIT – V - Financial and Non Financial Use cases of Blockchain:

Cross border payments, Know Your Customer , Blockchain in Healthcare, Blockchain in Energy Markets, Blockchain in Media, Preventing Cyber Crime through blockchain, Government Use-cases, Blockchain for Tax Payments, Blockchain for Managing Land Registry Records, Blockchains for e-voting.

Suggested Reading:

1) Second Edition, Mastering Bitcoin: Programming the Open Blockchain by Andreas M. Antonopoulos
2) Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

References:

1) Bitcoin and Cryptocurrency Technologies – A Comprehensive Introduction Hardcover, First Edition,by Arvind Narayanan Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, PRINCETON UNIVERSITY PRESS.
2) Blockchain – Blueprint for a New Economy by Melanie Swan
3) Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits - https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (M.Tech-I)

Course Code: CS 427 Title of the Course: Advanced Computer Architecture

L-T-P: 3-0-0 Credits: 3

Prerequisite Course / Knowledge (If any): Computer organizations, Basics of Computer Architecture, Operating Systems

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO1: Indicate how to estimate cycles taken for instructions with and without branch prediction, branch target buffer, etc. optimizations. (Understand)
- CO2: Compare the performance of the system with various optimizations such as dynamic scheduling and speculation versus regular pipelining. (Analyze)
- CO3: Summarise the problems involving the centralized and distributed shared-memory architectures and operations involved in accessing different memory locations present in multiple processors. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1				2		1	3		
CO2				1			3	2	
CO3					1		2	3	

Detailed Syllabus:

UNIT - I: Fundamentals of Quantitative Design and Analysis: Classes of computers, Trends in technology, Measuring and Reporting Performance, Power consumption and efficiency as the metric, Dependability, Quantitative Principles of Computer Design.

UNIT - II: Instruction Set Principles: Classifying Instruction Set Architectures, Memory Addressing, Addressing modes, Operations in the instruction set, Instructions for control flow, encoding an instruction set, Role of compilers.

UNIT - III: Memory Hierarchy Design: Cache performance review, Four basic memory hierarchy questions, Six basic optimizations of cache performance, Ten advanced optimizations of cache performance, Protection: virtual memory and virtual machines, memory technology and optimizations: SRAM, DRAM, Flash memory, Graphics Data RAMs.

UNIT - IV: Instruction-Level Parallelism and its Dynamic Exploitation: Instruction level parallelism: concepts and challenges, basics of pipelining, data hazards, structural hazards, control hazards, minimizing data hazards through forwarding, overcoming branch penalties by delayed branches, static and dynamic branch prediction, dealing with exceptions in pipelining, dynamic scheduling, Tomasulo's algorithm, speculative processors, high performance instruction delivery, VLIW approach, static and dynamic superscalar processors.

UNIT - V: Multiprocessors and Thread-Level Parallelism: Multithreading: exploiting thread-level parallelism within a processor, Symmetric shared-memory architectures and their performance, Distributed shared-memory architectures and their performance, synchronization, models of memory consistency.

Reference Books

1. John L. Hennessey and David A. Patterson. Computer Architecture: A Quantitative Approach, 5E, Morgan-Kaufmann, 2012. ISBN-13: 978-0123838728.

School of Computer & Information Sciences

Name of the Academic Program: M.Tech (M.Tech-I)

Course Code : CS428 Title of the Course : Graph Theory

L-T-P :. 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done discrete mathematics course.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

CO1: Describe the basic concepts of graphs (Understand)

CO3: Discuss the proofs of some properties of graphs (Understand)

CO4: Model real world problems using graph theory (Analyze)

CO5: Formulate and prove central theorems about trees, connectivity, planar graphs, coloring and matching. (Create)

CO6: Create basic algorithms of graphs (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
CO7												
CO8												

Detailed Syllabus:

UNIT-I: Introduction and Basic Concepts

Introduction to undirected/directed graphs, applications of graphs, basic definitions of graphs: graphic sequence, subgraph, paths, walks, cycles, Euler tour and Hamiltonian cycle, graph isomorphism, different kinds of graphs: bipartite, complete, regular graphs, etc.

UNIT-II: Trees

Properties of trees, centres, enumerating trees, fundamental circuits, spanning trees

UNIT-III: Graph connectivity

Vertex and edge connectivity, bi-connected components. Cut vertex and cut sets.

UNIT-IV: Planarity and graph Coloring

Euler formula for planar graphs, dual graphs, Kuratowski theorem. Vertex coloring, chromatic number, chromatic polynomial, coloring planar graphs: 5-coloring and 4-coloring planar graphs

UNIT-V: Other topics

Matching, maximum matchings, Halls matching condition, min-max theorem, maximum bipartite matching. Network flows, max flow min-cut theorem

Reference Books:

1. Douglas West, Introduction to Graph Theory, Pearson Education India.
2. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice Hall India.
3. Adrian Bondy and U.S.R. Murty, Graph Theory, Springer.

School of Computer & Information Sciences

M.Tech (Computer Science)

Scheme

II-Semester					
Core			Optional Core/Elective		
Code	**Course Title**	**Credits**	**Code**	**Course Title**	**Credits**
CS451	Software Engineering	4		System Security	
	Optional core-III	4		Network Security	
	Elective –II	¾		Virtualization	
	Elective –III	¾		Cloud Computing	
	Elective –IV	¾		Grid Computing	
CS452	*IT Lab –II*	2		Mobile Computing	
CS453	*SE Lab*	2		Wireless Sensor Networks	
CS454	*Communication Skills*	3		Advanced Computer Networks	
		21/24			
IV-Semester					
Code	**Course Title**	**Credits**			
	Project	12			

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 451 Title of the Course : **Software Engineering**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO1: Explain the models of software development process (Understand) CO2: Evaluate the appropriateness of different models of software development for their application in various domains (Evaluate).
- CO3: Apply the requirements engineering to software systems. (Apply)
- CO4: Describe Software Architectures (understand).
- CO5: Assess the applicability of software architectures for various combinations of non-functional requirements (Evaluate level).
- CO6: Apply object oriented and structured and structured paradigms to design software systems (Apply).
- CO7: Apply testing strategy to test software applications (Apply).

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			2	3	1				
CO2			3	2	1				
CO3			1	3	2				
CO4			3	2	1				
CO5	3		2		1				
CO6		1		3	2				
CO7	2	1		3					

Detailed Syllabus:

UNIT-I: Introduction to Software Engineering

Need of software engineering, systems engineering, challenges in software engineering, Software process models, quality characteristics of software systems, Ethics in Software Engineering.

UNIT-II: Requirements Engineering

Requirements engineering process, requirements specification, structured and object oriented analysis

UNIT-III: Software Design

Architectural design, detailed design, Structured and object oriented design, user interface design

UNIT-IV: Software Testing

Verification, Validation, testing techniques, Testing Process

UNIT-V: Tools and Evolution

CASE Tools, Reverse engineering, Reengineering and Configuration management.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition.

2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

4. Grady Booch, James Rumbaugh, Ivor Jacobson(2005), "*The Unified Modeling Language User Guide*", 2nd Edition, Addison Wesley Professional.US

School of Computer and Information Sciences

Name of the Academic Program: M.Tech (Computer Science) (M.Tech-II)

Course Code: CS452 Title of the Course: IT Lab-II (Web Technologies Lab)
L-T-P: 0-0-3 Credits : 2
Prerequisite Course / Knowledge (If any): --None

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO-1: Design web pages using scripting languages, cascading styles sheets and identify its elements and attributes. (Create)
- CO-2: Develop web pages using client side technologies and perform event handling and validation
- procedures. (Create)
- CO- 3: Create schemas and documents using markup languages, design and develop lightweight data-interchange formats for exchange of data between client and server applications. (Create)
- CO-4: Apply JavaScript libraries to create dynamic web page, access and use web services for interactive web contents. (Apply)
- CO-5: Develop applications using server side technologies, implement session management, database connectivity, and create dynamic HTML content with servlets and Java server pages. (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			3	2	1				
CO2			3	2	1				
CO3			2	3	1				
CO4			2	3	1				
CO5		1		2	3				

Detailed Syllabus:

UNIT- I: HTML, Forms & CSS

Introduction To HTML5, WWW, W3C, web publishing, Introduction To Style sheet, types of style sheets- Inline, External, Embedded CSS, text formatting properties, CSS Box Model, CSS Border, margin properties, Positioning Use of classes in CSS, color properties, use of &, Layout Design using CSS.

UNIT- II: JavaScript

Intro to script, types, intro of JavaScript, JavaScript identifiers, operators, control & Looping structure, Intro of Array, Array with methods, Math, String, Date Objects with methods User defined & Predefined functions, DOM objects, Window Navigator, History, Location, Event handling, Validations On Forms.

UNIT- III: Representing Web Data: XML, JSON

XML, XML Schema and DTD document definitions, XSLT transformations and programming, XPath, XQuery, Introduction to JSON.

UNIT- IV: jQuery& AJAX

Introduction to jQuery, Syntax Overview, Anatomy of a jQuery Script, Creating first jQuery script, Traversing the DOM, Selecting Elements with jQuery, Refining & Filtering Selections, Selecting Form Elements, Working with Selections, Chaining, Getters & Setters, CSS, Styling, & Dimensions, Manipulating Elements, Getting and Setting Information about Elements, Moving, Copying, and Removing Elements, Creating New Elements, Manipulating Attributes, Utility Methods, Events, Connecting Event to Elements, Namespacing Events, Event handling, Triggering Event handlers, Event Delegation, Animating effects, animate(), click(), hover(), toggle(), Plugins , Create a basic plugin, Finding &Evaluating Plugins, Writing Plugins, Tabs, Panels and Panes examples, jQuery UI and Forms, AJAX Overview, jQuery's AJAX related methods, Ajax and Forms, Ajax Events

UNIT- V: Java based Server Side Programming

Introduction to Java Programming Language, Features of JAVA; Java Language Basics, Packages and Interfaces, Exception handling; Database Management through JDBC, Connecting to a database for creation or manipulation; Multithreading Programming, Introduction to Servelet, HTTP Servelet Class, Request Interface, Response Interface, Session Tracking, Database Connectivity from Servelet, Interservelet Communication, Servelet Collaboration, Overview of JSP, Relation of Applets and Servelets with JSP, Scripting Elements, JSP Expressions, JSP Scriplets, JSP Declarations, Predefined Variables, Creating Custom JSP Tag Libraries Using Nested Tags, Structuring Generated Servelet in JSP Pages, Including Files and Applets in JSP Documents, Integrating Servelet and JSP.

Reference Books:

1. Internet and World Wide Web How to Program, P.J. Deitel, H.M. Deitel
2. Jeffrey C. Jackson, "Web Technologies—A Computer Science Perspective", Pearson Education, 2006.
3. Complete reference HTML.
4. JavaScript Bible
5. HTML, DHTML, JavaScript, Perl & CGI Ivan Bayross
6. XML: How to program Deitel&Deitel.

Suggested Exercises

1. Develop an Ajax application so that it uses any of the file (JSON or XML) as input and displays the read data without changing the front end of the application. (Assume appropriate members and data for the design and development of the required application).
2. Design and develop a graphical user interactive application with various components (elements). Also perform client-side validation using JavaScript.
3. Write JS code that reads XML file or JSON file and print the details as tabular data. (Assume appropriate members and data for the design and development of the required application).
4. Using JSP technologies develop an application to perform the following operations:: Login/Register, make a topic, leave replies, edit content, delete content. Create different permissions for different users – simple users should only be allowed to edit (not delete) the topics and replies that were created by them. Admins should be able to delete and edit anything.
5. Develop an application using HTML, CSS and Java Script such that access to JSON data from URL parameters will display the data based on the search keywords mentioned in the textbox. (Assume appropriate members and data for the design and development of the required application).
6. Using JSP technologies develop an interactive application to support selection, addition, deletion and searching operations.
7. Create a basic plugin for the operations using jQuery UI and Forms.
8. Create a document that reads and stores cookies containing a user name and number of times, he or she has visited your website. Whenever the user visits the site, the system displays the cookies in alert dialogue box, increments the counter cookie by 1 and then resets the counter's expiration date to one year from the current date.
9. Using Web Technologies to develop Visual Aids.
10. Practice exercise of developing web base application with the help of JSP and databases.
11. Design and develop GUI to perform the event handling operations and triggering the events.
12. Develop an application for exemplifying the use of Unobtrusive jQuery at Client Side.

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 453 Title of the Course: **Software Engineering Lab**

L-T-P : 0-0-3 Credits : 2

Prerequisite Course / Knowledge (If any): It is expected that the students must have done at least one programming course at undergraduate/postgraduate level

Course Outcomes (COs)

After completion of this course successfully, the students will be able to

- CO1: Create user stories (Create)
- CO2: Develop test plans for test first development (Create)
- CO3: Design & develop the stories (Create)
- CO4: Create the documentation(Create)
- CO5: Develop Software requirements specification document(Create)
- CO6: Apply object oriented and structured paradigm(Apply)
- CO7: Generate test reports (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1	2			3	1				
CO2		2	3	1					
CO3			2	3	1				
CO4	2	3			1				
CO5		1	3	2					
CO6		1		2	3				
CO7		2	3	1					

Detailed Syllabus:

For a given case study/problem statement, the following deliverables are to be realized

- Define stories
- Identify tasks and develop test plan for stories/task (with the help of specifications)
- Design and develop increments
- Test the increments and release the increment
- Apply object oriented and structured modelling
- Implement the case study for plan driven approach by writing use case specification, designing the system and implementing the same.

Reference Books:

1. Ian Sommerville (2016), "*Software Engineering*",10th Edition, Pearson Education Limited, Global Edition
2. Roger S Pressman, Bruce R Maxim(2015), "*Software Engineering, A Practitioner's Approach*", 8th Edition, TataMcGraw Hill, Indian Edition

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 472 Title of the Course: **Cloud Computing**

L-T-P : 4-0-0 Credits : 4

Prerequisite Course / Knowledge (If any): Knowledge of Distributed Computing, Networks, Java/C++/Python/C#/node.js and Unix/Linux/Windows.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Describe the fundamentals of Service Oriented Architecture and its implementations (Understand)
- CO2: Analyze a practical application, identify the issues and apply suitable techniques studied in the course effectively (Analyze)
- CO3: Develop skill to learn how to write Web Services using SOAP and REST way. (Create)
- CO4: Outline the theory behind the Virtualization of x86 family of processors (Understand)
- CO5: Apply the system virtualization solutions for creating Virtual Machine and their management (Apply)
- CO5: Illustrate Data Management approaches in Cloud and apply the solutions using HDFS/Spark (Apply)
- CO6: Assess the proposed solutions (REST/SOAP), architecture using different parameters for a given problem (Evaluate)
- CO7: Describe the subtle issue in handing the security in Clouds (Understand)
- CO8: Develop skill to learn how to write Cloud Services/deploy using PaaS solutions like (Aneka &Heroku). (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1				2			3	1	
CO2			3	2				1	
CO3				2			3	1	
CO4			2	3				1	
CO5				2	3			1	
CO6				2			3	1	
CO7			2	3				1	
CO8			2		3			1	

Detailed Syllabus:

UNIT- I. Basics of Web Services: Distributed computing using SOA, Extensible Mark-up Language XML Introduction, some key aspects of XML, XML-based Web Services, Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL), UDDI (Universal Description Discovery and Integration) discovery that form a basis for Web Services, exploring JAXR, jUDDI, UDDI4J etc. Representational state transfer (REST) software framework, Development of Java Web Services using SOAP and REST.

UNIT- II. Virtualization and Resource Provisioning in Clouds: Introduction to Cloud Technologies, Study of Hypervisors Virtualization Technology: Virtual machine technology, virtualization applications in enterprises, Pitfalls of virtualization

UNIT- III. Introduction to Cloud Computing: Definition, Characteristics, Components, Cloud provider, SAAS, PAAS, IAAS and Others, Organizational scenarios of clouds, Administering & Monitoring cloud services, benefits and limitations, Deploy application over cloud, Comparison among SAAS, PAAS, IAAS Cloud computing platforms: Infrastructure as service: Amazon EC2,Platform as Service examples: Google App Engine, Microsoft Azure, Heroku. Utility Computing, Elastic Computing.

UNIT- IV Data Management in Clouds: Data in the cloud: Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo. Map-Reduce and extensions: Parallel computing, Introduction to cloud development, Example/Application of MapReduce, Features and comparisons among GFS,HDFS etc, Map-Reduce model.

UNIT- V. Security in Clouds: Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud computing security architecture: Architectural Considerations- General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro-architectures; Identity Management and Access control-Identity management, Access control, Autonomic Security Cloud computing security challenges: Virtualization security management- virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud.

UNIT- VI. Programming Enterprise Clouds using Aneka: Introduction, Aneka Architecture, Aneka Deployment, Parallel Programming Models, Thread Programming using Aneka, Task Programming using Aneka, and MapReduce Programming using Aneka, Parallel Algorithms, Parallel Data mining, Parallel Mandelbrot, and Image Processing.

UNIT- VII. Advanced Topics and Cloud Applications: Cloud computing platforms, Installing cloud platforms and performance evaluation. Features and functions of cloud platforms: Xen Cloud Platform, Eucalyptus, OpenStack, Open Shift/Kubernetes.

Text books:

A. Books (Web Services SOAP and REST)

1. Henry Bequetet. al. (2002), *Beginning Java web services*, WROX Press Ltd; Illustrated edition, Programming web services with SOAP, James Snell et. al., O' Reilly publisher.
2. James Snell et. al.(2009), *Programming web services with SOAP*, Illustrated edition,O' Reilly publisher.
3. Leonard Richardson, Sam Ruby et al.(2007), "RESTful Web Services", first edition,O'Reilly Media.
4. George Reese (2012), *The REST API Design*, Kindle Edition, O'Reilly Media.
5. Mario-Leander Reimer (2018), *Building RESTful Web Services with Java EE 8: Create modern RESTful web services with the Java EE 8 API*, Kindle Edition Packt Publishing.

B. Books (Cloud)

1. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi(2013), *Mastering Cloud Computing: Foundations and Applications Programming,* first edition, Morgan Kaufmann.

2. Kai Hwang , Jack Dongarra, Geoffrey C. Fox. (2013), *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things,* first edition, Morgan Kaufmann.

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 473 Title of the Course: **System Security**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Computer Networks, Operating Systems.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO-1: Describe the concepts of system security (Understand)
- CO-2: Apply various security concepts while building a robust system (Apply)
- CO-3: Evaluate the security posture of an organization (Evaluate)
- CO-4: Analyze the security loopholes in a system/organization (Analyze)
- CO-5 : Create a security and privacy aware environment in the organization/society (Create)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			3	1					1
CO2				2	1				3
CO3		3			1				2
CO4	3					1			2
CO5	2				1				3

Detailed Syllabus:

UNIT-I: **An Overview of Computer Security**: Confidentiality, Integrity, Availability, Threats Policy and Mechanism, Goals of Security, Assumptions and Trust, Assurance, Specification, Design. Policies for security, confidentiality and integrity. Hybrid models

UNIT-II: **Authentication, Entities and Systems for Enforcement:** Authentication, Physical Security, Access Control systems, Biometrics, Identity, Trust

UNIT-III: **Security in Operating Systems:** Protected Objects and Methods of Protection, Secure Programs, Non-malicious Program Errors, Viruses and Other Malicious Code, Trusted Operating System Design, Secure Programming, File and Memory security, Attacks, Threats, kernel flaws and vulnerabilities. Backups and recovery, Hardening of systems, Robust Programming.

UNIT-IV: **Case Study of Operating System Security:** Windows Security: Security components of Windows OS, Windows registry, Windows domains, Active Directory, Access Controls, Principals, Subjects Tokens, Security Identi.ers, Privileges, Objects, Security Descriptors, ACE Matching, Microsoft Malware Protection Engine (MSMPENG), Baseline security analysis.
Unix/Linux Security: Types of attacks, Unix system security checklist, Integrity Management, Auditing and Logging, Protecting Against Programmed Threats, Wrappers and Proxies, SUID or SGID loopholes, Patching and updates to software. Android and other Mobile OS Security.

UNIT-V: **Legal Ethical, Privacy Issues Audits and Administration of Security:** Privacy Concepts, Privacy Principles and Policies, Computer Crime, Ethical Issues in Computer Use, Protecting Programs and Data, Forensics, Logging and Analysis, Security Planning, Risk Analysis, Organizational Security Policies and enforcement

Suggested Reading::

1. Ross J. Anderson (2008), *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley.
2. Charles P. P.eeger and Shari Lawrence P.eeger, *Security in Computing*, Pearson Education.
3. Matt Bishop, *Computer Security: Art and Science*, Pearson Education.
4. Simson Gar.nkel and Gene Spafford, *Practical UNIX & Internet Security*, 2nd Edition, O'Rielley.

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 475 Title of the Course: **Network Security**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Computer Networks, Operating Systems.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO1: Discuss various security policies and models used in network security (Understand).
- CO2: Review the Policy models for confidentiality, integrity and availability triad (CIA) in different scenarios with case studies. (Understand)
- CO3: Apply different security protocols across TCP/IP networks to protect networks against threats, vulnerabilities and attack vectors (Apply).
- CO4: Evaluate the need for cryptography and other state of the art technology their importance in modern electronic era in detecting, preventing and recovery (Evaluate).
- CO5: Examine how techniques, technology, can manage information and infrastructure protection in organizations using security development life cycle. (Analyze)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1		2	3						1
CO2		3	2						1
CO3				2	1				3
CO4	2					1			3
CO5		1			3				2

Detailed Syllabus:

PART I SECURITY CHALLENGES TO COMPUTER NETWORKS

1. **SECURITY THREATS TO COMPUTER NETWORKS** Sources of Security Threats, Security Threat Motives, Security Threat Management, Security Threat Correlation, Security Threat Awareness.
2. **COMPUTERNETWORKVULNERABILITIES** Sources of vulnerabilities, Vulnerability Assessment.
3. **HOSTILE SCRIPTS** Introduction to the Common Gateway Interface (CGI), CGI Scripts in a Three – Way Handshake, Server – CGI Interface, CGI Script Security Issues, Web Script Security Issues, Dealing with the Script Security Problems, Scripting languages: server side, client side scripting languages

PARTII DEALING WITH NETWORK SECURITY CHALLENGES

4. **ACCESSCONTROLANDAUTHORIZATION** Access Rights, Access Control Systems, Authorization, Types of Authorization Systems, Authorization Principles, Authorization Granularity, Web Access and Authorization.

5. **AUTHENTICATION**: Authentication Elements, Types of Authentication, Authentication Methods, Developing an Authentication Policy.

6. **FIREWALLS** Types of Firewalls, Configuration and Implementation of a Firewall, Demilitarized Zone (DMZ), Firewall Services and Limitations.

7. **SYSTEMINTRUSIONDETECTIONANDPREVENTION** Intrusion Detection, Intrusion Detection Systems, Types of Intrusion Detection Systems, Challenges toIntrusion Detection Systems, Intrusion Prevention Systems

PART III PROTOCOLS AND STANDARDS

8. NETWORKSECURITYPROTOCOLSANDSTANDARDS Application Level Security: PGP, S/MIME, Secure HTTP, HTTPS, Secure Electronic Transactions (SET), Kerboros; Security in the Transport Layer: Secure Socket Layer (SSL), Transport Layer Security (TLS); Security in the Network Layer: Internet Protocol Security (IPSec), Virtual Private Networks (VPNs), Security in the Link Layer and over LANS: Point-to-Point Protocol (PPP), Remote Authentication Dial-In Service (RADIUS), Terminal Access ControllerAccess Control System (TACACS+)

Suggested Reading:

1. William Stallings, Cryptography and Network Security: Principles and Practices, 6th Edition, Pearson Publication, 2013.

2. Matt Bishop, Computer Security, Art & Science Second Edition or higher, Addison Wesley; (11 September 2015)

Additional Reading

1. Network Security: Private Communications in a Public World (Radia Perlman Series in Computer Networking and Security) 2nd Edition

2. https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/.

3. RFCS & Journal papers

School of Computer and Information Sciences

Name of the Academic Program: **M.Tech (Computer Science) (M.Tech-II)**

Course Code: CS 476 Title of the Course: **Advanced Computer Networks**

L-T-P : 3-0-0 Credits : 3

Prerequisite Course / Knowledge (If any): Computer Networks course with a minimum grade of B. In general a thorough and complete understanding of TCP/IP stack and other general networking concepts is essential. A Good understanding of Operating Systems is also required.

Course Outcomes (COs)

After completion of this course successfully, the students will be able to:

- CO-1: Uunderstand the structure of RFCs and Concepts related to services and protocols in the IETF. (Understand).
- CO-2: Analyze the architecture of IPv6 networks and generalize concepts across IPv4 and IPV6. (Analyze)
- CO-3: Understand how complex lookups in a routing table can be avoided using MPLS. (Understand)
- CO-4: Apply the knowledge of TE, QoS and QoE to the design of high performance data networking (Apply)
- C0-5: Distinguish between traditional networks and software defined networks and also understand the benefits of network softwarization. (Understand)
- CO-6: Characterize the important aspects of networks applicable to Cloud and Data Centers through SDN, and NFV. (Understand)

Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Program Specific Outcomes (PSOs)

	PO1	PO2	PO3	PO4	PO5	PO6	PSO1	PSO2	PSO3
CO1			3	2			1		
CO2			2	3			1		
CO3				2	3		1		
CO4				2	3		1		
CO5			3	2			1		
CO6			2	3			1		

Detailed Syllabus:

The course will consist of areas that are some of the current trends in networking. The last part will be the latest papers from journals and conferences on the current topics in networking. Relevant RFCs from IETF will be used as references for all the units where applicable.

UNIT-I: IPv6: The next generation internet – this is the trend of the future and many other aspects of later sections will assume understanding of IPv6. The basic IPv6 protocol with its new auto-configuration schemes will be studied. The transition technologies for moving from IPv4 to IPv6 will also be studied.

UNIT-II: Mobile IP and Mobile IPv6: In this we study Mobile IPv4 (MIPv4) and Mobile IPv6 (MIPv6). We study the basic mobile IPv4 protocol and also the triangular and optimized routing. Then, we study the way mobile IPv6 protocol and how it is different from mobile IPv4. We also study the basic handoff mechanisms that are proposed to ensure low latency.

UNIT-III: Multiprotocol Label Switching (MPLS): Basics of label switching, Advantages, MPLS key concepts, MPLS control and data planes, advertising and using labels in MPLS, Label stacking, and Label Distribution Operations.

UNIT-IV: Traffic Engineering (TE)/Quality of Service (QOS) and QoE: Considering the latest trend towards VoIP, quality of service is of utmost importance in the Internet. We will study the basic concepts of QoS and the various proposals to achieve QoS in the Internet that have been standardized by the IETF such as Diffserv and Intserv and MPLS+Diffserv. User Quality of Experience (QoE): definition and strategies in practice.

UNIT-V: Data Center Networking: Data center networking has led to modifications to TCP which can be studied such as Multipath TCP (MPTCP), Data Center TCP (DCTCP) etc. and some newer models of congestion control.

UNIT-VI: Software Defined Networking (SDN) and Network Functions Virtualization (NFV): In this section, motivation for introducing SDNs is studied followed by understanding data plane abstraction, control plane abstraction and network virtualization concepts. OpenFlow as a south bound interface of SDN will be studied. NFV almost plays a role of equal importance to that of SDN in modern networking. So NFV concepts, benefits, requirements and reference architecture and NFV functionality are studied along with the relationship between SDN and NFV.

STUDENT SEMINARS: (optional) in this part, students will be required to do literature survey and study some of the latest papers in advanced topics of networking and give an in-depth seminar on the chosen topic.

TEXTBOOKS and REFERENCES

5. RFCs for IPv6 and Microsoft documents on IPv6. Here the relevant RFCs and their implementational aspects will be expected.
6. *Mobile IP, Design Principles and Practices* by Charles Perkins, Pearson Education (2008), **ISBN-10:** 8131720977, **ISBN-13:** 978-8131720974
7. *Mobile IPv6: Mobility in a Wireless Internet* by Hesham Soliman, Addison Wesley, **ISBN-10:** 0201788977, **ISBN-13:** 978-0201788976
8. *MPLS and Label Switching Networks* by Uyless Black, Prentice Hall, **ISBN-10:** 0130158232, **ISBN-13:**

978-0130158239

9. *MPLS: Technology and Applications* by Bruce Davie and Yakov Rekhter, Morgan Kaufmann, **ISBN-10:** 1558608885, **ISBN-13:** 978-1558608887
10. *QoS control in High Speed Networks* by H.Jonathan Chao, Xiaolei Guo, Wiley Interscience, **ISBN-10:** 9780471003977, **ISBN-13:** 978-0471003977
11. Software Defined Networking: A comprehensive approach, by Paul Goransson (Author), Chuck Black (Author), Timothy Culver (Author), ISBN-13: 978-0128045558, ISBN-10: 0128045558, Morgan-Kauffman.
12. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud by William Stallings, Pearson, ISBN-13: 978-0-13-417539-3, ISBN-10: 0-13-417539-5.
13. Research papers and Books for data center networking.
14. Cloud Native Data Center Networking: Architecture, Protocols, and Tools, Dinesh G Dutt, O'Reilly Media, ISBN-10: 1492045608, ISBN-13: 978-1492045601
15. Bob Lantz, Brandon Heller, and Nick McKeown. A Network in a Laptop: Rapid Prototyping for Software-Defined Networks. 9th ACM Workshop on Hot Topics in Networks, October 20-21, 2010, Monterey, CA.
16. http://mininet.org/walkthrough/ (to learn mininet commands and setup)
17. https://osrg.github.io/ryu/ (to know about ryu, a python based SDN controller)

Suggested Assignments and Case Studies

1. Some case studies such as Skype to understand how QoS is actually done in the Internet are studied.

2. mininet (an open source emulator) and ryu (an open source SDN controller) set up is explained and demonstrated to understand the SDN concepts. The above references points the material and links to understand mininet and ryu tools.

3. IPv6 functionality is demonstrated using mininet. For example, add 'a' as a ipv6 node in mininet then specify inet6 as the address family to retrieve the IPv6 information, like 'a ip -f inet6 link/address/route', 'ping6' is for IPv6 ping and 'traceroute6' is used to trace IPv6 path.

4. Using mininet and ryu, few SDN applications can be developed like firewall and load balancer.