CURRICULUM & SYLLABI

M.Tech. (CSE)

COMPUTER SCIENCE & ENGINEERING

Effective from AY: 2024-25



NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

WARANGAL, TELANGANA

Table of Contents

VISION AND MISSION OF THE INSTITUTE 3
VISION AND MISSION OF THE DEPARTMENT 3
PROGRAM EDUCATIONAL OBJECTIVES 4
PROGRAM OUTCOMES 5
CURRICULUM 6
SYLLABI 10

Vision and Mission of the Institute
National Institute of Technology Warangal

VISION

Towards a Global Knowledge Hub, striving continuously in pursuit of excellence in Education, Research, Entrepreneurship and Technological services to the society

MISSION

- Imparting total quality education to develop innovative, entrepreneurial and ethical future professionals fit for globally competitive environment.
- Allowing stake holders to share our reservoir of experience in education and knowledge for mutual enrichment in the field of technical education.
- Fostering product-oriented research for establishing a self-sustaining and wealth creating centre to serve the societal needs.

Vision and Mission of the Department
Computer Science & Engineering

VISION

Attaining global recognition in Computer Science & Engineering education, research and training to meet the growing needs of the industry and society.

MISSION

- MS1: Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry.
- MS2: Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science and engineering.
- MS3: Developing linkages with world class organizations to strengthen industry-academia relationships for mutual benefit.

Program: M.Tech. (Computer Science & Engineering)

Program Educational Objectives

PEO-1	Design, develop and test software systems for engineering applications.
PEO-2	Analyze technical solutions to computational problems and develop efficient algorithms.
PEO-3	Work in multi-disciplinary teams to specify software requirements and to achieve project goals.
PEO-4	Communicate effectively and demonstrate professional ethics with societal responsibilities.
PEO-5	Engage in lifelong learning to keep pace with changing landscape of technologies for professional advancement.

Program Articulation Matrix

PEO / Mission Statements	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
Imparting quality education through well-designed curriculum in tune with the challenging software needs of the industry	3	2	1	1	2
Providing state-of-art research facilities to generate knowledge and develop technologies in the thrust areas of computer science and engineering.	2	2	2	-	2
Developing linkages with world class organizations to strengthen industry- academia relationships for mutual benefit	2	2	2	2	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Program: M.Tech. (Computer Science & Engineering)

Program Outcomes

PO-1	Engage in critical thinking and pursue investigations / research and development to solve practical problems.
PO-2	Communicate effectively, write and present technical reports on complex engineering activities by interacting with the engineering fraternity and with society at large.
PO-3	Demonstrate higher level of professional skills to tackle multidisciplinary and complex problems related to Computer Science and Engineering.
PO-4	Apply concepts of theoretical computer science to design software systems satisfying realistic, economic, social, safety and security constraints.
PO-5	Design and develop processes to meet targeted needs with optimum utilization of resources.
PO-6	Develop robust, reliable, scalable techniques and tools for knowledge-based systems.

MAPPING OF PROGRAM OUTCOMES WITH PROGRAME EDUCATIONAL OBJECTIVES

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

CURRICULUM
M.Tech. (Computer Science & Engineering)

1st Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS16001	Advanced Algorithms	3-0-0	3
2	CS16003	Machine Learning	3-0-0	3
3	CS16005	Advanced Operating Systems	3-0-0	3
4	CS160XX	Professional Elective – I	3-0-0	3
5	CS160XX	Professional Elective – II	3-0-0	3
6	CS16007	Advanced Operating Systems Laboratory	0-1-2	2
7	CS16009	Machine Learning Laboratory	0-1-2	2
			Total Credits	**19**

2nd Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS16002	Advanced Computer Networks	3-0-0	3
2	CS16004	Advanced Software Engineering	3-0-0	3
3	CS160XX	Professional Elective - III	3-0-0	3
4	CS160XX	Professional Elective - IV	3-0-0	3
5	CS160XX	Professional Elective - V	3-0-0	3
6	CS16006	Advanced Computer Networks Laboratory	0-1-2	2
7	CS16008	Advanced Software Engineering Laboratory	0-1-2	2
8	CS16095	Minor Project	0-0-0	2
			Total Credits	**21**

3rd Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS16089	Seminar and Technical Writing	0-0-0	2
2	CS16091	Summer Internship/Research Experience	0-0-0	2
3	CS16093	Comprehensive Viva-Voce	0-0-0	2
4	CS16097	Dissertation: Part - A	0-0-0	8
			Total Credits	**14**

4th Semester

S.No.	Code	Course Title	L-T-P	Credits
1	CS16098	Dissertation: Part - B	0-0-0	**12**
			Total Credits	**12**

Professional Elective Courses:

Professional Elective-I & II		
S.No.	**Code**	**Course Title**
1	CS16021	Advanced Compiler Design
2	CS16023	Advanced Databases
3	CS16025	Big Data
4	CS16027	Bio-Informatics
5	CS16029	Cloud Computing
6	CS16031	Computer Vision & Image Processing
7	CS16033	Cryptography and Network Security
8	CS16035	Data Science Fundamentals
9	CS16037	Distributed Computing
10	CS16039	Game Theory
11	CS16041	Privacy Preserving Data Publishing
12	CS16043	Quantum Computing
13	CS16045	Software Reliability and Quality Management

Professional Elective-III, IV & V		
S.No.	**Code**	**Course Title**
1	CS16022	Advanced Data Mining
2	CS16024	Exploratory and Interactive Data Analysis
3	CS16026	Fault Tolerant Systems
4	CS16028	Fog and Edge Computing
5	CS16030	High Performance Computing
6	CS16032	Human Computer Interaction
7	CS16034	Information Retrieval
8	CS16036	Internet of Things
9	CS16038	Models in Deep Learning
10	CS16040	Models for Social Networks
11	CS16042	Natural Language Processing
12	CS16044	Randomized and Approximation Algorithms
13	CS16046	Real Time Systems
14	CS16048	Reinforcement Learning
15	CS16050	Security and Privacy
16	CS16052	Service Oriented Architecture and Micro-Services
17	CS16054	Social Media Analytics
18	CS16056	Soft Computing Techniques
19	CS16058	Software Defined Networks
20	CS16060	Vehicular Networks

SYLLABI

M.Tech. (Computer Science & Engineering)

1st Semester

CS16001 **3-0-0 (3)**

ADVANCED ALGORITHMS

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze worst-case running times of algorithms using asymptotic analysis.
CO-2	Classify problems into different complexity classes corresponding to deterministic, approximation and parameterized algorithms.
CO-3	Analyze the complexity of graph problems for different graph classes.
CO-4	Analyze approximation algorithms and determine approximation factor.
CO-5	Design and analyze efficient randomized algorithms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	-	3	3	-	-
CO-2	3	-	3	3	-	-
CO-3	3	-	3	3	-	-
CO-4	3	-	3	3	-	-
CO-5	3	-	3	3	-	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Average case analysis of algorithms, Correctness of Master Theorem, Selection in Worst Case Linear Time, Large integer multiplications using FFT, Dynamic Programming - Matrix Chain Multiplication Problem, Optimal Binary Search Tree, Linear Algorithm for Domination in Trees, Maximum Cardinality Search and Chordal Graphs, Greedy Algorithm for Optimal Coloring of Chordal Graphs, NP-completeness, Efficient Reduction Proofs via Examples, Domination in Subclasses of Bipartite Graphs and Chordal Graphs, Exact Exponential Algorithm for Domination Problems, Treewidth, Parameterized Complexity Classes, APX-hardness and APX-completeness, Approximation Algorithm for Connected Dominating Set Problem, The Stable Marriage Problem, The Coupon Collector`s Problem.

Learning Resources:

Text Books:

1. *Introduction to Algorithms*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI, 2009, Third Edition.
2. *Fundamentals of Computer Algorithms*, Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Universities Press, 2011, Second Edition.
3. *Algorithmic Graph Theory and Perfect Graphs*, Martin Charles Golumbic, 2004, Elsevier, Second Edition.

Reference Books:

1. *Parameterized Complexity*, Rodney G. Downey and M. R. Fellows, Springer, 2012.
2. *Approximation Algorithms*, Vijay V. Vajirani, Springer, 2001.
3. *Randomized Algorithms*, Rajeev Motwani and Prabhakar Raghavan, Cambridge University Press, 1995.

Other Suggested Readings:

1. *Treewidth: Computations and Approximations*, Ton Kloks, Springer-Verlag, 1994.
2. *Graph Classes A Survey* : Andreas Brandstädt, Van Bang Le and Jeremy P.Spinard, SIAM, 1987.
3. *Algorithms and Complexity*, Herbert S. Wilf, AK Peters/CRC Press, 2002, Second Edition.

CS16003 **3-0-0 (3)**

MACHINE LEARNING

Pre-Requisites: **None**

Course Outcomes:

CO-1	Construct prediction models for statistical data.
CO-2	Design Multi-Layer neural network to solve Supervised Learning problems.
CO-3	Classify non-linear data like face recognition, disease prediction.
CO-4	Apply Genetic Algorithm for optimization problems.
CO-5	Design applications like games and agent-based controllers.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	-	1	1	2	-
CO2	2	2	2	2	1	2
CO3	3	2	1	2	2	-
CO4	2	3	-	2	1	-
CO5	2	-	2	-	1	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to machine learning, issues related to machine learning: pre-processing, inductive bias, variance, feature extraction, feature selection techniques. Different types of learning, training and testing, hypothesis and cost function. Mathematics for machine learning. Classification techniques: DECISION TREES (DT)-construction of decision trees using different algorithms, Regression tree, tree pruning, rule extraction from trees, multivariate trees. Ensemble learning: Bagging and boosting and different techniques of bagging and boosting. Artificial neural networks (ANN): different learning rules, single-layer perceptron, multi-layer neural nets, backpropagation algorithm, feed-forward networks, network training, radial basis function networks, recurrent neural networks. Bayesian learning: probabilistic reasoning: prior, likelihood and posterior, belief networks: modeling independencies, Markov equivalence in belief networks, hidden Markov models (HMM).Naïve Bayes classifier, learning with hidden variables, Expectation Maximisation (EM). - GENETIC ALGORITHMS – an illustrative example, Hypothesis space search, Genetic Programming, Models of Evolution and Learning. Instance-based learning: Nearest-Neighbour classification, condensed-neighbour classification. Unsupervised linear dimensionality reduction: principal component analysis(PCA), PCA vs singular value decomposition, working on high-dimensional data, latent-semantic analysis: information retrieval. Supervised linear dimensionality reduction: Fisher's linear discriminant. Kernel methods: dual representations, kernel construction, learning with hyperparameters, support vector machines (maximum margin classifier), linear and multiclass SVMs. REINFORCEMENT LEARNING - The Learning Task, Q Learning, Nondeterministic rewards and actions, Temporal difference learning, Generalizing from examples, relationship to Dynamic Programming.

Learning Resources:

Text Books:

1. Tom M .Mitchell, Machine Learning, McGraw Hill, 1997.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

1. Stuart Russell and Peter Norvig, Artificial Intelligence, A Modern Approach, Pearson Publishers, 2020, 4th Edition.

CS16005 **3-0-0 (3)**

ADVANCED OPERATING SYSTEMS

Pre-Requisites: **None**

Course Outcomes:

CO-1	Design and implement Unix kernel data structures and algorithms.
CO-2	Analyze synchronization problems in uniprocessor and multiprocessor systems.
CO-3	Evaluate the scheduling requirements of different types of processes and find their solutions.
CO-4	Implement user level thread library and mimic the behavior of Unix kernel for scheduling, synchronization and signals.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2		3		3	-
CO-2	2		3		3	-
CO-3	2		3		3	-
CO-4	2		3		3	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to UNIX: The process and the kernel, Mode, space and context, Process abstraction, kernel mode, synchronization by blocking interrupts, process scheduling.
Introduction to Threads: Fundamental abstractions, Lightweight process design, Issues to consider, User level thread libraries, scheduler activations Signals: Signal generation and handling, Unreliable signals, Reliable signals, Signals in SVR4, Signals implementation, Exceptions, Process Groups
Process Scheduling: Clock interrupt handling, Scheduler Goals, Traditional UNIX scheduling, Scheduling case studies
Synchronization and Multiprocessing : Introduction, Synchronization in Traditional UNIX Kernels, Multiprocessor Systems, Multiprocessor synchronization issues, Semaphores, spin locks, condition variables, Read-write locks, Reference counts, Practice and solving problems on synchronization, process scheduling and threads.
Introduction to Intel X86 Protected Mode: Privilege Levels, Flat memory model, Descriptors - Segment, Task, Interrupt; GDT, LDT and IDT, Initializing to switch to protected mode operation, Processor Exceptions.
Kernel Memory Allocators: Resource map allocator, Simple power-of-two allocator, McKusick-Karels Allocator, Buddy system, SVR4 Lazy Buddy allocator, OSF/1 Zone Allocator, Hierarchical Allocator, Solaris Slab Allocator
File system interface and framework : The user interface to files, File systems, Special files, File system framework, The Vnode/Vfs architecture, Implementation Overview, File System dependent objects, Mounting a file system, Operations on files.
File System Implementations : System V file system (s5fs) implementation, Berkeley FFS, FFS functionality enhancements and analysis, Temporary file systems, Buffer cache and other special-purpose file systems Distributed File Systems : Network File System (NFS), Remote File Sharing (RFS) Advanced File Systems : Limitations of traditional file systems, Sun-FFS, Journaling approach 4.4 BSD, Log-Structured file system, Meta logging Episode FS, Watchdogs, 4.4 BSD portal FS, Stackable FS layers, 4.4 BSD FSinterface.

Learning Resources:

Text Books:

1. Thomas M. Cover and Uresh Vahalia, *UNIX Internals*, Pearson Education, 2005.
2. Richard Stevens, Stephen A. Rago, *Advanced Programming in the UNIX Environment*, Pearson

CS16007 **0-1-2 (2)**

ADVANCED OPERATING SYSTEMS LABORATORY

Pre-Requisites: **None**

Course Outcomes:

CO1	Implement basic/UNIX kernel level algorithms.
CO2	Implement the user level thread library and mimic the behavior of UNIX kernel for scheduling, synchronization and signals.
CO3	Implement File system image in a file and NFS using RPC.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	2	-	2	1	2	-
CO2	2	-	2	2	2	-
CO3	2	-	2	2	2	-

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Write Command Interpreter Programs which accepts some basic Unix commands and displays the appropriate result. Each student should write programs for at least six commands.
2. Study the concept of Signals and write a program for Context Switching between two processes using alarmsignals.
3. Study pthreads and implement the following: Write a program which shows the performance improvement in using threads as compared with process.(Examples like Matrix Multiplication, Hyper quicksort, Merge sort, Traveling Sales Person problem)
4. Create your own thread library, which has the features of pthread library by using appropriate system calls (UContext related calls). Containing functionality for creation, termination of threads with simple round robin scheduling algorithm and synchronization features.
5. Implement all CPU Scheduling Algorithms using your threadlibrary
6. Study the concept of Synchronization and implement the classical synchronization problems using Semaphores, Message queues and shared memory (minimum of 3 problems)
7. A complete file system implementation inside a disk imagefile.
8. NFS server and NFS client implementation usingRPC

Learning Resources:

Text Books:

1. *Richard Stevens, Stephen A. Rago, Advanced Programming in the UNIX Environment, Pearson Education, 2/e, 2005.*

CS16009 **0-1-2 (2)**

MACHINE LEARNING LABORATORY

Pre-Requisites: **None**

Course Outcomes:

CO-1	Implement data cleaning and extraction.
CO-2	Implement prediction and classification algorithms on real-world data.
CO-3	Apply optimization techniques.
CO-4	Improve classification and regression algorithms using ensemble learning.
CO-5	Design and develop application specific neural networks.
CO-6	Design and develop gaming applications using reinforcement learning.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	3	3	2	3
CO-2	3	1	3	3	3	3
CO-3	3	1	3	3	3	3
CO-4	3	2	3	3	3	3
CO-5	3	3	3	3	3	3
CO-6	3	3	3	3	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

1. Data preprocessing, feature extraction and selection techniques.
2. Linear and logistic regression techniques.
3. Decision trees, regression trees
4. Ensemble techniques-bagging and boosting
5. Optimization techniques-genetic algorithms and particle swarm optimization
6. Bayesian networks, K-nearest algorithm
7. Unsupervised learning algorithms-clustering, dimensionality reduction
8. Support vector machine.
9. Artificial neural networks-multilayer perceptron networks.
10. Reinforcement learning

Note: All the above experiments will be implemented without libraries and next inbuilt libraries on real-world datasets.

Learning Resources:

Text Books:

1. Tom M .Mitchell, Machine Learning, McGraw Hill, 1997.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

2. Stuart Russell and Peter Norvig, Artificial Intelligence, A Modern Approach, Pearson Publishers, 2020, 4th Edition.

2nd Semester

CS16002 **3-0-0 (3)**

ADVANCED COMPUTER NETWORKS

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO-1	Analyze computer network architectures and estimate quality of service.
CO-2	Design application-level protocols for emerging networks.
CO-3	Analyze TCP and UDP traffic in data networks.
CO-4	Design and Analyze medium access methods, routing algorithms and IPv6 protocol for data networks.
CO-5	Analyze Data Center Networks and Optical Networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	2	3	2	–
CO-2	2	1	2	2	–	–
CO-3	1	–	2	2	–	–
CO-4	1	–	2	2	1	–
CO-5	1	–	2	2	1	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Application requirements, Network Architecture, Performance: Bandwidth and Latency, High Speed Networks

Connecting to a network: Framing, Error Detection, Reliable Transmission, Ethernet and Multiple Access Networks, Client-Server Networks, Wireless Networks, Delay-Tolerant Networks

Internetworking: Switching: Circuit-Switched Networks, Datagram Networks, Virtual-Circuit Networks, Message-Switched Networks, Asynchronous Transfer Mode, Exploring Broadband Integrated Services Digital Network, Optical networking: SONET/SDH Standards, Learning bridges, IPv4: Address Space, Notations, Classful, Classless, Datagram, Fragmentation and Checksum, ICMP, ARP, RARP, Network Address Translation; IPv6 Addresses: Structure, Address Space, Packet Format and Extension Headers, Intra-Domain and Inter-Domain Routing, Unicast Routing Protocols: RIP, OSPF and BGP, IGMP,Multicast Routing Protocols: DVMRP, PIM-DM, PIM-SM, CBT, MSDP and MOSPF, Spanning Tree Algorithm

Congestion Control and Resource Allocation: Problem, Issues, Queuing, TCP Congestion Control, Congestion-Avoidance Mechanisms and Quality of Service

Traffic Engineering: Requirement, Traffic Sizing, Characteristics, Protocols, Time and Delay Considerations, Connectivity, Availability, Reliability, Maintainability and Throughput, Software-Defined Traffic Engineering

Applications and Security: Multimedia Over Internet: Transmission, IP Multicasting and VoIP, Domain Name System: Name Space, Domain Name Space, Distribution, Domains, Resolutions and Dynamic Domain Name System, SNMP, Overlay Networks: Routing Overlays, Peer-to-Peer Networks and Content Distribution Networks, Datacenter Design and Interconnection Networks, Virtualization and Software-Defined Networking, Security: IPSec, SSL/TLS, PGP and Firewalls

Learning Resources:

Text books/ Reference books

1. Larry L. Peterson and Bruce S. Davie, Computer Networks: A System Approach, Sixth Edition, Morgan Kaufmann, Elsevier, 2020.
2. Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill, Sixth Edition, 2022.
3. Chwan-Hwa (John) Wu, J. David Irwin, Introduction to Computer Networks and Cyber Security, CRC press, Taylor & Francis Group,2016.
4. Andrew S. Tanenbaum, David J. Wetherall, Computer Networks, Pearson, 6th Edition, 2022.
5. G. Wright and W. Stevens, TCP/IP Illustrated, Volume 1 and Volume 2, Addison-Wesley, 1996.
6. Dayanand Ambawade, Deven Shah, Mahendra Mehra and Mayank Agarwal, Advanced Computer Network, Dreamtech Press, 2016.
7. R. Srikant, The Mathematics of Internet Congestion Control, Springer, 2004.
8. J. L. Boudec and P. Thiran, Network Calculus, Springer, 2011.

CS16004 **3-0-0 (3)**

ADVANCED SOFTWARE ENGINEERING

Pre-Requisites:

Course Outcomes:

Course Outcomes:

CO-1	Apply the Object Oriented Software-Development Process to design software.
CO-2	Design large-scale, reusable and complex software systems with Design and Architectural patterns.
CO-3	Develop and apply testing strategies for software applications.
CO-4	Analyze different Software Reliability parameters using Markovian Models, Finite Failure.
CO-5	Design and Plan software solutions to security problems using various paradigms.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	2	2	2	2
CO-2	2	3	2	2	2	2
CO-3	2	3	2	2	2	2
CO-4	2	3	2	2	2	3
CO-5	2	3	2	2	2	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction and System Engineering: Introduction, Software Process and Methodology, System Engineering. Analysis and Architectural Design: Software Requirement Elicitation, Domain Modeling, Architectural Design. Modeling and Design of Interactive Systems and Other Types of Systems: Deriving Use Cases from Requirements, Actor-System Interaction Modeling, Object Interaction Modeling, Applying Responsibility-Assignment Patterns, Deriving a design class diagram, User Interface Design. Object State modeling of Event-Driven Systems, Activity Modeling for Transformational Systems. Implementation and Quality Assurance: Implementation Considerations, Software Quality Assurance, Software Testing.
Software Reliability Modeling: Markovian Models, Finite Failure Category Models, Infinite Failure Category Models. Comparison of Software Reliability Models.
Project Management and Software Security: Software Project Management, Software Security

Learning Resources:

Text Books / Reference Books / Other Suggested Readings:

1. Kung, David. Object-oriented software engineering: an agile unified methodology. McGraw-Hill Higher Education, 2013.
2. Gamma, Erich. Design patterns: elements of reusable object-oriented software, Pearson Education India, 1995.
3. M. Xie, Software Reliability Modelling, World Scientific; 1991.
4. John D. Musa, Anthony Iannino, Kazuhira Okumoto, Software

CS16006 **0-1-2 (2)**

ADVANCED COMPUTER NETWORKS LABORATORY

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO-1	Develop programs for client-server applications.
CO-2	Perform packet sniffing and analyze packets in network traffic.
CO-3	Simulate and study the behavior of different network protocols.
CO-4	Implement network security algorithms.
CO-5	Develop a basic Software Defined Networking application .

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	1	2	3	2	–
CO-2	2	1	2	2	–	–
CO-3	1	–	2	2	–	–
CO-4	1	–	2	2	1	–
CO-5	1	–	2	2	1	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Assignment 1: Implementation of client server programs with different networking constraints, hosts behind NAT, hosts are globally addressable, etc

Assignment 2: Study and analysis of the network using Wireshark network protocol analyzer

Assignment 3: Generate different topologies for network simulation using NS3 and analyze its performance

Assignment 4: Simulate and study the behavior of different transport layer protocols using NS3

Assignment 5: Case-study for cloud / fog-based simulation tool

Assignment 6: Develop a basic Software Defined Networking application

Assignment 7: Configure different application servers such as web server, DNS servers, and email server, and secure using firewall rules

Learning Resources:

Text books/ Reference books

1. Larry L. Peterson and Bruce S. Davie, Computer Networks: A System Approach, Sixth Edition, Morgan Kaufmann, Elsevier, 2020.
2. G. Wright and W. Stevens, TCP/IP Illustrated, Volume 1 and Volume 2, Addison-Wesley, 1996.
3. W. Richard Stevens, UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998
4. ns-3 | a discrete-event network simulator for internet systems, online: https://www.nsnam.org/
5. CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services, online: https://github.com/Cloudslab/cloudsim
6. iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments, online: https://github.com/Cloudslab/iFogSim

CS16008 **0-1-2 (2)**

ADVANCED SOFTWARE ENGINEERING LABORATORY

Pre-Requisites:

Course Outcomes:

Course Outcomes:

CO-1	Proficient in identifying and categorizing requirements from problem statements.
CO-2	Skilled in estimating project metrics using advanced techniques like COCOMO.
CO-3	Competent in modeling UML use case diagrams and capturing diverse scenarios.
CO-4	Able to apply E-R modeling techniques effectively to represent data structures.
CO-5	Capable of designing comprehensive test suites and conducting software testing.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	2	2	2	2
CO-2	2	3	2	2	2	2
CO-3	2	3	2	2	2	2
CO-4	2	3	2	2	2	3
CO-5	2	3	2	2	2	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Identifying the Requirements from Problem Statements,

Requirements, Characteristics of Requirements, Categorization of Requirements, Functional Requirements, Identifying Functional Requirements,

Estimation of Project Metrics,

Project Estimation Techniques, COCOMO, Basic COCOMO Model, Intermediate COCOMO Model, Complete COCOMO Model, Advantages of COCOMO, Drawbacks of COCOMO, Halstead's Complexity Metrics,

Modeling UML Use Case Diagrams and Capturing Use Case Scenarios,

Use case diagrams, Actor, Use Case, Subject, Graphical Representation, Association between Actors and Use Cases, Use Case Relationships, Include Relationship, Extend Relationship, Generalization Relationship, Identifying Actors, Identifying Use cases, Guidelines for drawing Use Case diagrams,

E-R Modeling from the Problem Statements,

Entity Relationship Model, Entity Set and Relationship Set, Attributes of Entity, Keys, Weak Entity, Entity Generalization and Specialization, Mapping Cardinalities, ER Diagram, Graphical Notations for ER Diagram, Importance of ER modeling,

Identifying Domain Classes from the Problem Statements,

Domain Class, Traditional Techniques for Identification of Classes, Grammatical Approach Using Nouns, Advantages, Disadvantages, Using Generalization, Using Subclasses, Steps to Identify Domain Classes from Problem Statement, Advanced Concepts,

Statechart and Activity Modeling,

Statechart Diagrams, Building Blocks of a Statechart Diagram, State, Transition, Action, Guidelines for drawing Statechart Diagrams, Activity Diagrams, Components of an Activity Diagram, Activity, Flow, Decision, Merge, Fork, Join, Note, Partition, A Simple Example, Guidelines for drawing an Activity Diagram,

Modeling UML Class Diagrams and Sequence diagrams,

Structural and Behavioral aspects, Class diagram, Elements in class diagram, Class, Relationships, Sequence diagram, Elements in sequence diagram, Object, Life-line bar, Messages,

Modeling Data Flow Diagrams,

Data Flow Diagram, Graphical notations for Data Flow Diagram, Explanation of Symbols used in DFD, Context diagram and leveling DFD,

Estimation of Test Coverage Metrics and Structural Complexity,

Control Flow Graph, Terminologies, McCabe's Cyclomatic Complexity, Computing Cyclomatic Complexity, Optimum Value of Cyclomatic Complexity, Merits, Demerits,

Designing Test Suites,

Software Testing, Standards for Software Test Documentation, Testing Frameworks, Need for Software Testing, Test Cases and Test Suite, Types of Software Testing, Unit Testing, Integration Testing, System Testing, Example, Some Remarks

Learning Resources:

Text Books / Reference Books / Other Suggested Readings:

1. Rational Online Documentation
2. Booch, Jackobson and Rambaugh, "*UML Guide*" , Pearson Edu, 1999
3. IEEE Standards for SRS Documents, IEEE Std. 830.
4. Fenton NE, "*Software Metrics: A Rigorous Approach*", Chapman and Hall, 1991

CS16095 **0-0-0 (2)**

MINOR PROJECT

Pre-Requisites: **None**

Course Outcomes:

CO-1	Identifying the problem of a research project through literature survey.
CO-2	Analyze the given solution and reproduce it for better experience.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	2	2	2	1
CO-2	2	3	2	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

3rd Semester

CS16089 **0-0-0 (2)**

SEMINAR AND TECHNICAL WRITING

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze the selected topic, organize the content and communication to audience in an effective manner.
CO-2	Practice the learning by self study.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	1	1	–	–
CO-2	2	3	1	1	–	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

In Seminar and Technical Writing, every student is expected to prepare a well-organized report based on one / all of the following:

- by attending at least 5 expert lectures/ invited talks/ Seminar/ Popular lectures etc. organized by the institute/any of the departments, ideally in a specific domain or with the same theme.
- prepare a business or marketing plan based on patent search

The student is expected to consolidate the ideas from these lectures/patent searches and may even include material from other sources to strengthen the content of the report. The student should prepare a well-organized report based on the above and present it to the panel constituted by the department, for evaluation.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria.

Criteria	Description	Weightages
I	Clarity on the topic	
II	List of lectures attended	
III	Report	
IV	Presentation	
V	Response to questions	

Evaluation Criteria-CO Mapping

CO / Criteria	CO1	CO2
I	X	
II	X	
III		X
IV		
V		

CS16091 **0-0-0 (2)**

SUMMER INTERNSHIP/RESEARCH EXPERIENCE

Pre-Requisites: **None**

Course Outcomes:

CO-1	Analyze the practical application of all courses in post graduate curriculum of Computer Science and Engineering.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	3	3	3	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Every student has to undergo either a Summer Internship / Research project. The summer internship may be undergone in an Industry/Research organization. The research project shall be registered under the guidance of any faculty member in the institute. The student is required to submit a report and present the work before an evaluation committee, nominated by the Head of the Department.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Relevance of the area of work	15 Marks
II	Performance of the Task	15 Marks
III	Crucial learnings from the work	20 Marks
IV	Report Preparation	25 Marks
V	Presentation and Response to questions	25 Marks

Evaluation Criteria-CO Mapping

CO / Criteria	CO1
I	X
II	X
III	X
IV	X
V	X

CS16093 **0-0-0 (2)**

COMPREHENSIVE VIVA-VOCE

Pre-Requisites: **None**

Course Outcomes:

CO-1	Comprehend and correlate the understanding of all courses in post graduate curriculum of Computer Science and Engineering.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	3	3	3	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

In Comprehensive viva-voce each student will be evaluated for their overall comprehension of the course work and laboratory training that they have undergone. The students will be expected to answer questions orally, write down simple equations, draw plots, schematics, write simple code etc. as questioned by the panel. Assessment will be done by the panel based on the student response.

CS16097 **0-0-0 (8)**

DISSERTATION : PART-A

Pre-Requisites: None

Course Outcomes:

CO-1	Identify the problem of a research project through literature survey.
CO-2	Analyze the technical feasibility of the project.
CO-3	Propose a solution for the research problem.
CO-4	Analyze, design and implement the proposed solution using software engineering practices.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	2	2	2	1
CO-2	2	3	2	2	2	1
CO-3	2	3	2	2	2	1
CO-4	2	3	2	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Students are expected to choose real-world contemporary problem and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel will decide the suitability and worthiness of the project

Dissertation Evaluation:

i. The dissertation shall be submitted as per the schedule given in the academic calendar.
ii. The dissertation supervisor will periodically review the progress of the student and finally give his/her assessment of the work done by the student.
iii. The Dissertation Part – A will be evaluated for 100 marks, with the following weightages:

Sub-component	**Weightage**
a) Periodic evaluation by Guide	40 marks
b) Mid-term review	20 marks
c) End Semester viva-voce examination	40 marks

The midterm review and the end semester viva-voce examination will be conducted by a committee constituted by the Head of the Department.

Evaluation Criteria

The student will be evaluated by the panel based on the below criteria. Weightage for each criterion will be determined by the panel and will be informed to the students.

Criteria	Description	Weightages
I	Selection of Topic	
II	Literature Survey	
III	Defining the Objectives and Solution Methodology	
IV	Performance of the Task	

Evaluation Criteria - CO Mapping:

CO / Criteria	CO1	CO2	CO3	CO4
I	X			
II		X		
III			X	
IV				X

Refer to MTech. – Regulations for any further information regarding Mid-term review, End Sem evaluation, Template for report preparation and plagiarism.

4th Semester

CS16098 **0-0-0 (12)**

DISSERTATION: PART-B

Pre-Requisites: **None**

Course Outcomes:

CO-1	Synthesize and apply prior knowledge to designing and implementing solutions to open-ended computational problems while considering multiple realistic constraints.
CO-2	Design and Develop the software with software engineering practices and standards.
CO-3	Analyze Database, Network and Application Design methods.
CO-4	Evaluate the solution through various validation and verification methods.
CO-5	Practice CASE tools for solving software engineering CASE Studies.
CO-6	Analyze professional issues, including ethical, legal and security issues, related to computing projects.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	-	2	2	2	1
CO-2	2	-	2	2	2	1
CO-3	2	-	2	2	2	1
CO-4	-	-	2	2	2	1
CO-5	-	-	2	2	2	1
CO-6	-	-	2	2	2	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Description:

Students are expected to choose real-world contemporary problem and apply the engineering principles learned, to solve the problem through building prototypes or simulations or writing codes or establishing processes/synthesis/correlations etc. The department constituted panel will decide the suitability and worthiness of the project.

Dissertation Evaluation:

i. The dissertation shall be submitted as per the schedule given in the academic calendar.
ii. The dissertation supervisor will periodically review the progress of the student and finally give his/her assessment of the work done by the student.
iii. The Dissertation Part – B will be evaluated for 100 marks, with the following weightages:

Sub-component	Weightage
a) Periodic evaluation by Guide	40 marks
b) Mid-term review	20 marks
c) End Semester viva-voce examination	40 marks

The midterm review will be conducted by a committee constituted by the Head of the Department. The end semester examination will be conducted by an External Examiner along with the evaluation committee constituted by the Head of the Department.

Evaluation Criteria:

The student will be evaluated by the panel based on the below criteria as a continuation of Dissertation Part A. Weightage for each criterion will be determined by the panel and will be informed to the students.

Task	Description	Weightages (%)
IV	Performance of the Task	20
V	Dissertation Preparation	25
VI	Review (Presentation & Understanding)	30
VII	Viva-Voce	15
VIII	Publications /Possibility of publication	10

Evaluation Criteria-CO Mapping:

CO / Criteria	CO1	CO2	CO3	CO4	CO5	CO6
IV	X	X				
V				X		
VI			X			
VII				X		
VIII					X	X

Refer to M.Tech. – Regulations for any further information regarding Mid-term review, End Sem evaluation, Template for report preparation and plagiarism.

Professional Electives

CS16021 **3-0-0 (3)**

Advanced Compiler Design

Pre-Requisites: None

Course Outcomes:

CO-1	Understand code generation methods.
CO-2	Apply scalar variable optimizations and procedural optimizations on intermediate code.
CO-3	Apply machine level optimizations on the low level intermediate code.
CO-4	Perform loop restructuring transformations.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	1	1	-	-
CO-2	2	-	2	2	2	-
CO-3	2	-	2	2	2	-
CO-4	2	-	2	2	2	-

1 – Slightly; 2 – Moderately; 3 – Substantially

Syllabus:

Review of compiler fundamentals – lexical analysis, parsing, semantic analysis and intermediate code generation, error recovery, run time storage management, code generation. Code optimization – Peephole optimization, control flow analysis, data flow analysis, dependence analysis, redundancy elimination, loop optimization, procedural and inter procedural optimization, instruction scheduling. Compiling for High performance architectures, Compiling for scalar pipeline, compiling for vector pipeline, super scaler and VLIW processors, compiling for multiple issue processors, compiling for memory hierarchy. Parallelization and Vectorization, Dependence and dependence testing. Loop Normalization, Induction variable Exposure, Enhancing Fine Grained Parallelism, Loop Interchange, Scalar Expansion, Scalar and Array Renaming, Node splitting, Index-set splitting, Loop skewing.

Learning Resources:

Text Books:

1. Randy Allen, Kennedy, Optimizing Compilers for Modern Architectures: A dependence-based approach, Morgan Kaufmann Publishers, 2001
2. Steven S. Muchnick, Advanced Compiler Design and implementation, Morgan Kaufmann Publishers, 1997
3. Keith D. Cooper & Linda Torczon, Engineering a Compiler, Morgan Kaufmann, 2004.

CS16023 **3-0-0 (3)**

Advanced Databases

Pre-Requisites: None

Course Outcomes:

CO-1	Design distributed database for application development.
CO-2	Apply query optimization principles for optimizing query performance in centralized and distributed database systems.
CO-3	Design distributed database schema using principles of fragmentation and allocation.
CO-4	Apply distributed transaction principles for handling transactions in distributed database applications.
CO-5	Apply distributed database administration principles for managing distributed database.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	2	1	1	2
CO-2	2	-	3	1	2	2
CO-3	2	-	3	1	-	2
CO-4	2	-	3	2	-	2
CO-5	2	-	3	2	2	2

1 – Slightly; **2 – Moderately;** **3 – Substantially**

Syllabus:

Distributed Databases: Introduction to Distributed Database Systems, Distributed Database System Architecture; Top-Down Approach, Distributed Database Design Issues, Fragmentation, Allocation, Database Integration, Bottom-up approach, Schema Matching, Schema Integration, Schema Mapping; Data and Access Control, View Management, Data Security; Query processing problem, Objectives of Query processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing; Query Decomposition, Normalization, Analysis, Elimination of Redundancy and Rewriting; Localization of Distributed Data, Reduction for primary Horizontal, Vertical, derived Fragmentation; Distributed Query Execution, Query Optimization, Join Ordering, Static& Dynamic Approach, Semi-joins, Hybrid Approach; Taxonomy of Concurrency control Mechanisms, Lock-Based Concurrency Control, Timestamp-Based Concurrency Control, Optimistic Concurrency Control, Deadlock Management; Heterogeneity issues Advanced Transaction Models, Distributed systems 2PC& 3PC protocols, Replication protocols, Replication and Failures, HotSpares;
Parallel Databases: Introduction to Parallel Databases, Parallel Database System Architectures, Parallel Data Placement, Full Partitioning; Parallel Query Processing, Query Parallelism; Parallel Query Optimization, Search Space, Cost Model, Search Strategy; Load Balancing.

Learning Resources:

Text Books:

1. M T Ozsu, Patrick Valduriez, Principles of Distributed Database Systems, Prentice Hall, 1999.
2. S. Ceri and G. Pelaggati, Distributed Database System Principles and Systems, MGH, 1985.

CS16025 **3-0-0 (3)**

Big Data

Pre-Requisites: None

Course Outcomes:

CO-1	Understand big data challenges in different domains including social media, transportation, finance and medicine.
CO-2	Analyze the capability of No-SQL systems.
CO-3	Apply machine learning algorithms for data analytics.
CO-4	Analyze MAP-REDUCE programming model for better optimization.
CO-5	Analyze the capability of Stream Data Processing Systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	3	1	1	1
CO-2	1	-	3	2	1	2
CO-3	1	-	3	3	1	2
CO-4	1	-	3	2	2	3
CO-5	-	-	1	1	2	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Overview of Big Data Analytics: Big Data Characteristics, Big Data Challenges, Big Data Analytics system Architecture, Big Data Storage Technologies, Big Data Analytics Life Cycle; No SQL databases: Introduction to No SQL databases, Types of No SQL databases, CAP Theorem, Replication & Shading, MongoDB basics; Overview of Hadoop, Hadoop Ecosystem, Reading and Writing Files on Hadoop Distributed File System. Map Reduce basics, Map Reduce Algorithm Design. Graph Algorithms, Data Mining with Big Data, Stream Data Processing, Stream Data Processing System and Tools, Overview of Spark, Spark SQL-Mllib-GraphX, Apache Kafka, Big Data Analytics in Industry Verticals, Operationalizing Basic Data Analytic Methods Using R, Analytics for Unstructured Data.

Learning Resources:

Text Books:

1. Jure Leskovec, Anand Rajaraman, J D Ullman, Mining Massive Datasets, Cambridge University Press, 2nd Edition, 2014.
2. Jimmy Lin and Chris Dyer, Data Intensive Text Processing with Map Reduce, Morgan & Claypool Publishers,1st Edition, 2010.

Reference Books:

1. Bill Franks, Taming The Big Data Tidal Wave, 1st Edition, Wiley, 2012.
2. Johannes Ledolter, Data Mining and Business Analytics with R, Wiley, 2013.

CS16027 **3-0-0 (3)**

Bio Informatics

Pre-Requisites: None

Course Outcomes:

CO-1	Classify models used in bioinformatics.
CO-2	Compute homologues, analyze sequences, construct and interpret evolutionary trees.
CO-3	Analyze protein sequences to retrieve protein structures from databases.
CO-4	Design of biological data model.
CO-5	Apply homology modeling and computational drug design.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	-	-	2	-	-	2
CO-2	2	-	2	-	-	2
CO-3	2	-	2	-	-	2
CO-4	2	-	2	-	-	1
CO-5	2	-	2	-	-	-

1 – Slightly; **2 – Moderately;** **3 – Substantially**

Syllabus:

Introduction and Biological databases - Introduction, Sequence Alignment - Pair wise sequence alignment, Database similarity searching, Multiple sequence alignment, Profiles and hidden markov models, Molecular Phylogenetics - Phylogenetics basics, Phylogenetic Tree Construction Methods and Programs, Genomics and Proteomics - Genome mapping, assembly and comparison, Functional genomics, Proteomics, Structural Bioinformatics - Basics of protein structure, Protein structure prediction.

Learning Resources:

Text Books:

1. Jin Xiong, Essential Bioinformatics, 1th Edition, Cambridge University Press,2011.
2. Arthur M Lesk, Introduction to Bioinformatics, 2nd Edition, Oxford University Press,2007.

CS16029 **3-0-0 (3)**

Cloud Computing

Pre-Requisites: None

Course Outcomes:

CO-1	Understand cloud computing concepts and history.
CO-2	Mastering principles of parallel and distributed computing.
CO-3	Proficiency in virtualization techniques.
CO-4	Analyzing cloud computing architectures.
CO-5	Familiarity with industry cloud platforms.
CO-6	Exploring advanced cloud computing topics and tools.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	3	2	–	2	–
CO-2	3	3	2	2	2	1
CO-3	3	3	2	3	2	2
CO-4	3	3	2	2	2	1
CO-5	–	–	3	–	2	2
CO-6	–	–	3	–	3	2

1 – Slightly; 2 – Moderately; 3 – Substantially

Syllabus:

Introduction: Cloud computing at a glance; **Historical developments; Building cloud computing environments**

Principles of Parallel and Distributed Computing: Eras of computing; Parallel vs. distributed computing; Elements of parallel computing; Elements of distributed computing; Technologies for distributed computing

Virtualization: Introduction; Characteristics of virtualized environments; Taxonomy of virtualization techniques; Virtualization and cloud computing; Pros and cons of virtualization; Technology examples

Cloud Computing Architecture: Introduction; The cloud reference model; Types of clouds; Economics of cloud; Open challenges

Cloud Platforms in Industry: Amazon web services; Google AppEngine; Microsoft Azure

Cloud Applications: Scientific applications; business and consumer applications

Advanced Topics in Cloud Computing: Energy efficiency in clouds; Market-based management of clouds; Federated clouds/Intercloud; Third-party cloud services; Resource allocation; Task scheduling; Service management; Data management; Resource management; Security and privacy; Edge computing; Fog computing; Osmotic computing

Toolkits: CloudAnalyst; CloudSim; iFogSim; Haizea – An open source VM-based lease manager

Learning Resources:

Text Books:

1. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, Morgan Kaufmann, 2013

Reference Books:

1. Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, 2011
2. Tim Mather, Subra Kumaraswamy and Shahed Latif, Cloud Security and Privacy, O'Reilly, 2009

Other Suggested Readings:

1. NPTEL Course on Cloud Computing by Prof. Soumya Kanti Ghosh, IIT Kharagpur, Prof. Rajiv Mishra, IIT Patna, Prof. Bidisha Chaudhuri, Prof. Amit Prakash, IIIT Bangalore

CS16031 **3-0-0 (3)**

Computer Vision & Image Processing

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO1	Classify Image representations.
CO2	Apply Image transformation methods.
CO3	Implement image processing algorithms.
CO4	Design face detection and recognition algorithms.
CO5	Recover the information, knowledge about the objects in the scene.

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6
CO1	1					
CO2	2		2			1
CO3	2		2			1
CO4	2		2	2		1
CO5	2		2	2		2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

The image model and acquisition, image shape, sampling, intensity images, color images, range images, image capture, scanners. Statistical and spatial operations, Gray level transformations, histogram equalization, multi image operations. Spatially dependent transformations, templates and convolution, window operations, directional smoothing, other smoothing techniques. Segmentation and Edge detection, region operations, Basic edge detection, second order detection, crack edge detection, edge following, gradient operators, compass & Laplace operators. Morphological and other area operations, basic morphological operations, opening and closing operations, area operations, morphological transformations. Image compression: Types and requirements, statistical compression, spatial compression, contour coding, quantizing compression. Representation and Description, Object Recognition, 3-D vision and Geometry, Digital Watermarking. Texture Analysis.

Learning Resources:

Text Books

1. D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, PHI Learning, 2009.
2. Milan Soanka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision, Cengage Learning, 2014
3. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Pearson Education, 2007.

CS16033 **3-0-0 (3)**

Cryptography and Network Security

Pre-Requisites: None

Course Outcomes:

CO-1	Understand encryption algorithms and their significance in modern systems.
CO-2	Understanding of number theory required for cryptographic algorithms.
CO-3	Understand cryptographic algorithms to build secure protocols.
CO-4	Design of secure protocols to solve real-world scenarios.
CO-5	Apply foot printing, scanning, enumeration, and similar techniques to discover network and system vulnerabilities.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	1	–	–	–	–	–	–	–	2	3	–	–	–	–	-	–
CO-2	–	–	–	3	–	–	–	–	–	–	2	–	1	–	–	–
CO-3	–	1	–	–	2	–	–	–	–	–	–	–	–	3	–	–
CO-4	–	–	1	–	–	–	–	2	–	-	–	–	–	–	–	3
CO-5	–	–	–	–	–	1	2	–	–	–	–	–	–	–	3	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Basic Cryptography- Security Services, Security Attacks, Basic cryptography methods, Classical Encryption Techniques, Substitution Ciphers, Transposition Ciphers, time pad, Cryptanalysis Security attacks, Symmetric cryptography algorithms- DES, 3DES, AES, Public-key cryptography,

Number Theory- Introduction to number theory – Modular Arithmetic; Finite fields; Number theory properties – Primality testing; Fetmat's and Euler's theorem; Chinese remainder theorem; Integer factorization; discrete logarithm.

Public Key Cryptography: RSA, ECC, Generalized ElGamal Public Key Cryptosystem, Key Management and Distribution: Symmetric Key Distribution, Diffie Hellman key exchange, Distribution of Public Keys, X.509 Certificates Functionalities of entity, content authentication; message digests and hashing schemes; Key management and Distribution-Certificate authorities; PKI; MAC; Hashing; Digital Signatures-Authentication protocols; Digital Signature Standard.

Network Security: IP Security, Transport-Level Security, Electronic Mail Security, Firewalls- Firewall Properties; Design of firewalls; VPN's; Filtering; Intrusion detection system

Learning Resources:

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson, 2017,
2. Dan Boneh and Victor Shoup, A Graduate Course in Applied Cryptography, Stanford University 4th edition, 2017

Reference Books:

3. Jonathan Katz, Yehuda Lindell, Introduction To Modern Cryptography, Taylor & Francis Ltd, 2020,
4. Douglas R. Stinson Maura B. Paterson, Cryptography Theory and Practice, CRC Press, Fourth edition, 2004

CS16035 **3-0-0 (3)**

Data Science Fundamentals

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO-1	Apply statistical methods to data for inferences.
CO-2	Analyze data using Classification, Graphical and computational methods.
CO-3	Understand Data Wrangling approaches.
CO-4	Perform descriptive analytics and data visualization over massive data.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	3	2	-	3
CO-2	2	2	3	2	-	3
CO-3	2	2	3	2	-	3
CO-4	2	2	3	2	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Overview of Random variables and distributions.
Statistical learning: Assessing model accuracy, Bias-Variance Trade-Off, Descriptive Statistics, Dependent and Independent events; Linear Regression: Simple and multiple linear regressions, Comparison of Linear regression with K-nearest neighbors. Simple Hypothesis Testing, Student's t-test, paired t and U test, correlation and covariance, tests for association; association rules and correlations; PCA and SVD.
Classification: Linear and Logistic Regression, Bayesian Learning, LDA, QDA, K-Nearest Neigbhour, and comparison of classification methods.
Data Visualization and Graphical Analysis: Visualized exploratory data Analysis, Histograms and frequency polygons, Box-plots, Quartiles, Scatter Plots, Heat Maps. Matrix visualization, Scientific Design Choices in Data Visualization, Higher-dimensional Displays and Special Structures, Visual data mining.
Data Wrangling: Data Acquisition, Data Formats, Imputation, split-apply-combine paradigm.
Descriptive Analytics: Data Warehousing and OLAP, Data Summarization, Data de- duplication, Data Visualization using CUBEs.

Learning Resources:

Text books

1. Gareth James Daniela Witten Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, February 11, 2013, web link: www.statlearning.com (1 to 4chapters)
2. Mark Gardener, *Beginning R The statistical Programming Language*, Wiley,2015.
3. Han , Kamber, and J Pei, Data Mining Concepts and Techniques, 3rd edition, Morgan Kaufman, 2012. (Chapter 2 and Chapter4)

Reference books

1. Chun-houh Chen, Wolfgang Hardle, Antony Unwin, Handbook of Data Visualization, Springer, 2008

Online Resources:

1. https://www.kdnuggets.com/topic/data-science
2. https://www.kdnuggets.com/topic/data-visualization

CS16037 **3-0-0 (3)**

Distributed Computing

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO-1	Identify models of distributed computing.
CO-2	Analyze algorithms for coordination, communication, and synchronization in distributed systems.
CO-3	Design and implement distributed, fault-tolerant storage systems.
CO-4	Design and Implement distributed analytics using MapReduce.
CO-5	Design distributed algorithms for handling deadlocks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2				–
CO-2	2	1	2	2	–	–
CO-3	2	1	2	1	2	–
CO-4	2	1	2	1	2	–
CO-5	1	–	2	1	1	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Types of distributed systems, synchronous vs. asynchronous execution, design issues and challenges, model of distributed computation

Logical Time: Logical clocks, scalar time, vector time, efficient implementation of vector clocks, virtual time

Message ordering and group communication: Message ordering paradigms, group communication, causal order (CO), total order, propagation trees for multicast, publish-subscribe model

Global state and snapshot recording algorithms: Chandy-Lamport algorithm and its variations, snapshot algorithms for non-FIFO channels, snapshots in a \ causal delivery system, monitoring global state, necessary and sufficient conditions for consistent global snapshots, finding consistent global snapshots in a distributed computation

Distributed mutual exclusion algorithms: Lamport's algorithm, Ricart–Agrawala algorithm, Singhal's dynamic information-structure algorithm, Maekawa's algorithm, Agarwal–El Abbadi quorum-based algorithm, Raymond's tree-based algorithm

Deadlock detection in distributed systems: System model, Knapp's classification of distributed deadlock detection algorithms, Mitchell and Merritt's algorithm for the single resource model, Chandy– Misra–Haas algorithm for the AND and the OR model, Termination detection algorithm

Distributed Data Storage and Computation: Memory consistency models, replication, CAP theorem, distributed shared memory, distributed file systems, distributed transactions, eventual consistency, conflict-free replicated data types, failures, byzantine fault-tolerance, consensus protocols, distributed data analytics using MapReduce

Learning Resources:

Text books/ Reference books

1. Ajay D. Kshemakalyani and Mukesh Singhal, Distributed Computing, Cambridge University Press, 2008.
2. Maarten Van Steen and Andrew S. Tanenbaum, Distributed Systems - Principles and Paradigms, 4th edition, online: https://www.distributed-systems.net/index.php/books/ds4/, 2024.

CS16039 **3-0-0 (3)**

Game Theory

Pre-Requisites: None

Course Outcomes:

CO-1	Analyze games based on complete and incomplete information about the players.
CO-2	Analyze games where players cooperate.
CO-3	Compute Nash equilibrium.
CO-4	Apply game theory to model network traffic.
CO-5	Analyze auctions using game theory.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	2	1	1	1	–
CO-2	2	3	2	2	3	–
CO-3	2	1	3	2	2	1
CO-4	2	3	2	1	2	1
CO-5	2	3	2	3	3	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Noncooperative Game Theory: Games in Normal Form - Preferences and utility, examples of normal-form, Analyzing games: Pareto optimality, Nash equilibrium, Maxmin and minmax strategies, dominated strategies, Rationalizability, Correlated equilibrium

Computing Solution Concepts of Normal-Form Games: Computing Nash equilibria of two-player, zero-sum games, Computing Nash equilibria of two-player, general-sum games, Complexity of computing Nash equilibrium, Lemke–Howson algorithm, Searching the space of supports, Computing Nash equilibria of n-player, general-sum games, Computing maxmin and minmax strategies for two-player, general-sum games, Computing correlated equilibria

Games with the Extensive Form: Perfect-information extensive-form games, Subgame-perfect equilibrium, Computing equilibria, Imperfect-information extensive-form games, Sequential equilibrium

Other Representations: Repeated games: Finitely repeated games, Infinitely repeated games, automata, Stochastic games Bayesian games: Computing equilibria

CoalitionalGameTheory: Transferable Utility, Analyzing Coalitional Games, Shapley Value, the Core

Mechanism Design: Strategic voting, unrestricted preferences, Implementation, quasilinear setting, efficient mechanisms, and Computational applications of mechanism design, Task scheduling, Bandwidth allocation in computer networks

Auctions: Single-good auctions, Canonical auction families, Bayesian mechanisms, Multiunit auctions, combinatorial auctions

Learning Resources:

Text Books:

1. Noam Nisan, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, “Algorithmic Game Theory”, Cambridge University Press, 2007, First Edition.
2. Ronald Cohn Jesse Russell, Algorithmic Game Theory, VSD Publishers, 2012, First Edition.

Other Suggested Readings:

1. https://onlinecourses.nptel.ac.in/noc22_cs116/preview
2. https://ocw.mit.edu/courses/14-126-game-theory-spring-2016/

CS16041 **3-0-0 (3)**

Privacy Preserving Data Publishing

Pre-Requisites: None

Course Outcomes:

CO-1	Apply anonymization methods for sensitive data protection.
CO-2	Apply state-of-art techniques for data privacy protection.
CO-3	Design privacy preserving algorithms for real-world applications.
CO-4	Identify security and privacy issues in OLAP systems.
CO-5	Apply information metrics for Maximizing the preservation of information in the anonymization process.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	-	3	-	2
CO-2	2	-	-	3	-	2
CO-3	2	-	-	3	-	2
CO-4	2	-	-	3	-	2
CO-5	2	-	-	3	-	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Fundamentals of defining privacy and developing efficient algorithms for enforcing privacy, challenges in developing privacy preserving algorithms in real-world applications, privacy issues, privacy models, anonymization operations, information metrics, Anonymization methods for the transaction data, trajectory data, social networks data, and textual data, Collaborative Anonymization, Access control of outsourced data, Use of Fragmentation and Encryption to Protect Data Privacy, Security and Privacy in OLAP systems, Extended Data publishing Scenarios, Anonymization for Data Mining, publishing social science data, continuous user activity monitoring (like in search logs, location traces, energy monitoring), social networks, recommendation engines and targeted advertising.

Learning Resources:

Text Books:

1. Benjamin C.M. Fung, Ke Wang, Ada Wai-Chee Fu and Philip S. Yu, *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*, 1st Edition, Chapman & Hall/CRC, 2010.
2. Charu C. Aggarwal, *Privacy-Preserving Data Mining: Models and Algorithms*, 1st Edition, Springer, 2008.

CS16043 **3-0-0 (3)**

Quantum Computing

Pre-Requisites: None

Course Outcomes:

CO-1	Understand quantum computing principles.
CO-2	Design and analyze quantum circuits and algorithms.
CO-3	Implement quantum circuits and algorithms using Qiskit.
CO-4	Understand fundamentals of quantum communication.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	–	1	1	–	1
CO-2	2	–	2	2	3	2
CO-3	2	–	2	2	1	2
CO-4	3	–	2	2	2	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Preliminaries: Review of linear algebra and complex numbers.

Quantum Computation: Introduction to qubits, Multiple qubits, Dirac notation, Bloch sphere, Reversable Gates, Basic single- qubit gates, Two-qubit gates, Measurements, Quantum circuits, Bell state circuit, No-cloning theorem, Teleportation, Amplitude Amplification, Superdense coding, Physical realizations of qubits and Qiskit.

Quantum Algorithms: Introduction to query complexity, Deutsch algorithm, Deutsch-Josza algorithm, Bernstein-Vajirani algorithm, Simon`s algorithm, Quantum Fourier Transform, Quantum Phase Estimation, Grover`s Search Algorithm, Ordering finding using phase estimation, Shor`s algorithm, Quantum key distribution, BB84 protocol and HHL algorithm.

Learning Resources:

Text Books:

1. Michael A. Nielsen and Issac L. Chuang, Quantum Computation and Quantum Information,. Cambridge, 2010, 10th Anniversary Edition.
2. Noson S. Yanofsky and Mirco A. Mannucci, Quantum Computing for Computer Scientists, Cambridge University Press, 2008, 1st Edition.

Reference Books:

1. Eleanor G. Rieffel and Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, MIT Press, 2011.
2. Quantum Computing in Practice with Qiskit(R) and IBM Quantum Experience(R): Practical recipes for quantum computer coding at the gate and algorithm level with Python, Hassi Norlén, Packt Publishing, 2020.

Other Suggested Readings:

1. IBM Quantum Learning Courses: https://learning.quantum.ibm.com/ .
2. YouTube Link: Introduction to Quantum Computing Complete Course - Quantum Soar.

CS16045 **3-0-0 (3)**

SOFTWARE RELIABILITY AND QUALITY MANAGEMENT

Pre-Requisites: Software Engineering, Probability and Statistics

Course Outcomes:

CO-1	Understand Software Reliability during different phases of Software Development Life Cycle.
CO-2	Analyze Software Reliability parameters using Markovian Modeling.
CO-3	Estimate Software Reliability parameters using Maximum Likelihood and Least Square Method.
CO-4	Evaluate performance of Binomial-Type, Poison-Type and Markovian Models.
CO-5	Predict Software Reliability using Intelligent Techniques.
CO-6	Design Quality Attributes for Software Quality Assurance (SQA).

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	3	2	2	–
CO-2	3	2	3	1	2	–
CO-3	3	2	3	2	2	–
CO-4	3	2	2	2	2	–
CO-5	3	2	3	2	2	–
CO-6	3	2	3	2	2	–

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Software Reliability: The need for Software Reliability, Some Basic Concepts, Software Reliability and Hardware Reliability, Availability, Modelling and General Model Characteristics.

Software Reliability Modeling: Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric, Error Seeding Models, Failure Rate Models, Curve Fitting Models, Reliability Growth Models, Markov Structure Models, Time Series Models, Non-homogeneous Poison Process Models.

Markovian Models: General Concepts, General Poison-Type Models, Binomial -Type Models, Poison-Type Models, Comparison of Binomial-Type and Poison-Type Models, Fault Reduction Factor for Poison-Type Models.

Descriptions of Specific Models: Finite Failure Category Models, Infinite Failure Category Models. Parameter Estimation: Maximum Likelihood Estimation, Least Squares Estimation, Bayesian Inference. Comparison of Software Reliability Models: Comparison Criteria, Comparison of Predictive Validity of Model Groups, Evaluation of other Criteria.

Software Reliability Prediction: Problems associated with different Software Reliability Models, Software Reliability prediction parameters, Intelligent Techniques for Software Reliability Prediction.

Software Quality Management: Software Quality Attributes, Quality Measurement & Metrics, Verification & Validation Techniques, Verification & Validation in the Life Cycle, Software Quality Assurance functions, Tool support for SQA.

Learning Resources:

Text Books:

1. M. Xie, Software Reliability Modelling, World Scientific; 1991.
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, Software Reliability Measurement, Prediction, Application. McGraw-Hill Book Company; 1987.
3. Hoang Pham, System Software Reliability, Springer; 2005
4. David C. Kung, Object-Oriented Software Engineering: An Agile Unified Methodology, McGrawHill Education (India) Edition 2015.

CS16022 **3-0-0 (3)**

Advanced Data Mining

Pre-Requisites: nil

Course Outcomes:

CO-1	Analyze Algorithms for frequent itemsets and sequential patterns.
CO-2	Determine patterns from time series data.
CO-3	Develop algorithms for Temporal Patterns.
CO-4	Apply Graph mining algorithms to Web Mining.
CO-5	Analyze algorithms for trajectory patterns.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	-	2	3	-	3
CO-2	3	-	2	3	-	3
CO-3	2	-	2	3	-	3
CO-4	2	-	2	3	-	3
CO-5	3	-	2	3	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Association Rule Mining, Sequential Pattern Mining concepts, primitives, scalable methods; Transactional Patterns and other temporal based frequent patterns, Mining Time series Data, Periodicity Analysis for time related sequence data, Trend analysis, Similarity search in Time-series analysis; Graph Mining, Mining frequent subgraphs, finding clusters in large graphs; Web Mining, Mining the web page layout structure, mining web link structure, Classification of web documents and web usage mining; Trajectory Pattern Mining: Moving together patterns, Sequential Pattern mining from trajectories.

Learning Resources:

Text Books:

3. **Jiawei Han and M Kamber**, Data Mining Concepts and techniques**, Morgan Kaufmann Publishers In, 2022; Fourth Edition.**
4. Chris Chatfield, The Analysis of Time Series: An Introduction, Chapman & Hall/CRC, 2003, Sixth Edition.
5. Bing Liu, Web Data Mining, Springer, 2011, Second Edition.

Reference Books:

3. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar, Introduction to Data Mining, Pearson, 2018, Second Edition.
4. G. Dong and J Pei, Sequence Data Mining, Springer, 2007, First Edition.

CS16024 **3-0-0 (3)**

Exploratory and Interactive Data Analysis

Pre-Requisites: Basic knowledge of statistics and programming such as Python or R.

Course Outcomes:

CO-1	Equipped the students with necessary skills to analyze and interpret data effectively.
CO-2	Students able to collect, visualize and summarize the data using software tools and libraries.
CO-3	Use data exploration and visualization techniques for multivariate and time series data.
CO-4	Students able to handle data with high dimensionality.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	2	3	2	1	2
CO-2	3	3	3	2	1	2
CO-3	3	3	3	2	1	2
CO-4	2	2	2	1	1	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction and Tools: Exploratory Data Analysis- Definition and importance, Role of EDA, EDA vs. Confirmatory Data Analysis. Tools and Technologies- Python, R, Jupyter Notebooks, etc..

Data Collection and Preparation: Data Sources and Collection-Databases, APIs, Web Scraping. Importing Data in Python (pandas) and R. Data Cleaning-Identifying and Handling Missing Data, Removing Duplicates, Correcting Data Errors, Data Types and Conversions. Data Transformation- Scaling and Normalization, Encoding Categorical Variables, Feature Engineering.

Data Visualization: Importance and types of Data Visualizations. Tools and Libraries: Python-Matplotlib, R-ggplot2, etc. Visualizing Univariate Data- Histograms, Box Plots, Bar Charts. Visualizing Multivariate Data-Scatter Plots, Pair Plots, Heatmaps. Advanced Visualizations- Interactive Visualizations, Geospatial Data Visualization.

Data Summarization: Descriptive Statistics- Measures of Central Tendency (Mean, Median, Mode), Measures of Dispersion (Range, Variance, Standard Deviation), Skewness and Kurtosis. Data Summarization- Grouping and Aggregation, Pivot Tables. Exploring Relationships: Correlation Analysis, Covariance.

Time Series Analysis and Dimensionality Reduction: Time Series Analysis- Time Series Decomposition, Trend and Seasonality. Anomaly Detection- Outliers, Anomaly Detection. Dimensionality Reduction-PCA, ICA, SVD, etc.

Hypothesis Testing and Statistical Analysis: Null and Alternative Hypotheses, Type I and Type II Errors. Common Statistical Tests- t-tests, Chi-square Tests, ANOVA.

Reporting and Dashboarding: Creating Effective Reports, Storytelling with Data. Dashboarding-Introduction to Dashboards, Tools for Dashboarding: Tableau, Power BI, etc.

Reference Books:

1. Suresh Kumar Mukhiya, Usman Ahmed, “Hands-On Exploratory Data Analysis with Python”, Packt Publishing, 2020.
2. Jake Vander Plas, "Python Data Science Handbook: Essential Tools for Working with Data", First Edition, O Reilly, 2017.
3. Catherine Marsh, Jane Elliott, “Exploring Data: An Introduction to Data Analysis for Social Scientists”, Wiley Publications, 2nd Edition, 2008.
4. Claus O. Wilke, “Fundamentals of Data Visualization”, O’reilly publications, 2019.
5. Matthew O. Ward, Georges Grinstein, Daniel Keim, “Interactive Data Visualization: Foundations, Techniques, and Applications”, 2nd Edition, CRC press, 2015.

CS16026 **3-0-0 (3)**

Fault Tolerant Systems

Pre-Requisites: None

Course Outcomes: At the end of the course, the student will be able to

CO-1	Understand the risk of computer failures and their comparison with other equipment failures.
CO-2	Know the different advantages and limits of fault avoidance and fault tolerance techniques.
CO-3	Analyze cost dependability trade-offs and the limits of computer system dependability.
CO-4	Gain knowledge in sources of faults and their prevention and forecasting.
CO-5	Analyze fault-tolerant or non-fault-tolerant based on dependability requirements.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	–	3	3	–	–
CO-2	3	–	3	3	–	–
CO-3	3	–	3	3	–	–
CO-4	3	–	3	3	–	–
CO-5	3	–	3	3	–	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Linear Introduction to Fault Tolerant Systems, Fault Classification, Types of Redundancy, Traditional and Network Measures, Hardware Fault Tolerance, Rate of Hardware Failures, Failure Rate, Reliability, Availability, Mean Time to Failure, Canonical and Resilient Structures: Series and Parallel Systems, Non-Series/Parallel Systems, M of N Systems, Triple Modular Redundant Structure, Voters, Variations on N-Modular Redundancy and Duplex Systems. Fault Tolerant Processor Level Techniques: Watchdog Processor and Simultaneous Multithreading, Byzantine Failures. Introduction to Information Redundancy, Coding: Parity, Checksum, M of N, Berger, Cyclic, Arithmetic, Resilient Disk Systems: RAID Level 1, Level 2, Level 3, Level 4 and Level 5, Data Replication: Non-Hierarchical, Hierarchical Organization and Primary-Backup Approach, Algorithm Based Fault Tolerance. Introduction to Software Fault Tolerance, Acceptance Tests, Single Version Fault Tolerance: Wrappers, Software Rejuvenation, Data Diversity and Software Implementable Hardware Fault Tolerance, N Version Programming: Consistent Comparison Problem and Version Independence, Recovery Block Approach, Preconditions, Postconditions and Assertions, Exception Handling, Software Reliability Models: Jelinski-Moranda, Littlewood-Verrall, Musa-Okumoto and Schneidewind Model, Remote Procedure Cells. Introduction to Checkpointing, Level, Latency, Overhead, Optimal Checkpointing: Reducing Overhead and Reducing Latency, Cache Aided Rollback Error Recovery, Checkpointing in Distributed Systems: Domino Effect and Livelock, Coordinated Checkpointing Algorithm, Time Based Synchronization, Diskless Checkpointing and Message Logging, Checkpointing in Shared Memory Systems: Bus-Based Coherence and Directory-Based Protocol, Checkpointing in Real Time Systems, Other Uses of Checkpointing.

Learning Resources:

Text Books/Reference Books/Online Resources:

1. I. Koren and C. M. Krishna, Fault Tolerant Systems, Morgan Kauffman, 2007.

2. D. K. Pradhan, Editor, Fault Tolerant Computer System Design, Prentice Hall, 1996.
3. L. L. Pullum, Software Fault Tolerance Techniques and Implementation, Artech House Computer Security Series, 2001.
4. M. L. Shooman, Reliability of Computer Systems and Networks Fault Tolerance Analysis and Design, Wiley, 2002.
5. Authors, Title of the Text Book, Name of the Publisher, Year of publication, Edition

CS16028 **3-0-0 (3)**

FOG & EDGE COMPUTING

Pre-Requisites: None

Course Outcomes: At the end of the course, the student will be able to

CO-1	Understand the basic requirements of fog and edge computing.
CO-2	Understand the key architectures and applications in fog and edge computing.
CO-3	Perform fog and edge computing services.
CO-4	Implement software using standard open-source fog and edge computing software for data analytics.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	3	2	-	3
CO-2	2	2	3	2	-	3
CO-3	2	2	3	2	-	3
CO-4	2	2	3	2	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Fog Computing, Limitation of Cloud Computing, Differences between Cloud and Fog Computing, Advantages, Business Models, Architecture, Opportunities and Challenges, Challenges in Fog Resources: Taxonomy and Characteristics, Resource Management Challenge, Optimization Challenges, Miscellaneous Challenges, IoT and Fog: Programming Paradigms, Research Challenges and Research Directions, Fog Protocols, Management and Orchestration of Network Slices in 5G, Fog, Edge and Clouds, Data Management and Analysis in Fog Computing, Case Studies. Introduction to Edge Computing, Origins of Edge, Edge Helping Low-End IoT Nodes, Architecture, Edge Helping Higher-Capability Mobile Devices: Mobile Offloading, Edge Helping the Cloud, Edge for Augmented Reality, Data Processing on the Edge, Dispersed Learning with Edge/Fog Computing, Video Analytics on the Edge, Edge Computing Applications.

Learning Resources:

Text Books:

1. Rajkumar Buyya, Satish Narayana Srirama, Fog and Edge Computing, Wiley Publications, 2019.
2. Wei Change and Jie Wu, Fog/Edge Computing for Security, Privacy and Applications, Springer, 2021.

CS16030 **3-0-0 (3)**

HIGH PERFORMANCE COMPUTING

Pre-Requisites: None

Course Outcomes: At the end of the course, the student will be able to

CO-1	Design and analyze the parallel algorithms for real world problems and implement them on available parallel computer systems.
CO-2	Optimize the performance of a parallel program to suit a particular platform.
CO-3	Design algorithms suited for Multicore processor systems using OpenCL, OpenMP, Threading techniques.
CO-4	Analyze the communication overhead of interconnection networks and modify the algorithms to meet the requirements.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	2	3	3	2	2
CO-2	3	2	3	3	2	2
CO-3	3	3	3	2	2	2
CO-4	2	3	3	2	2	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Implicit parallelism, Limitations of memory system performance, control structure, communication model, physical organization, and communication costs of parallel platforms, Routing mechanisms for interconnection networks, mapping techniques.

Parallel algorithm design: Preliminaries, decomposition techniques, tasks and interactions, mapping techniques for load balancing, methods for reducing interaction overheads, parallel algorithm models.

Basic communication operations: Meaning of all-to-all, all-reduce, scatter, and gather, circular shift and splitting routing messages in parts.

Analytical modeling of parallel programs: Sources of overhead, performance metrics, the effect of granularity on performance, scalability of parallel systems, minimum execution time, minimum cost-optimal execution time, asymptotic analysis of parallel programs.

Programming using message passing paradigm: Principles, building blocks, MPI, Topologies and embedding, Overlapping communication and computation, collective communication operations, Groups and communicators.

Programming shared address space platforms: Threads, POSIX threads, Synchronization primitives, attributes of threads, mutex and condition variables, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, An overview of Intel Threading building blocks.

Basic parallel algorithms: Prefix sums, Tree traversal algorithms, basic operations (insertion deletion and search) on trees, merging, maximum, graph colouring list ranking, Planar geometry and String algorithms

Dense Matrix Algorithms: Matrix vector multiplication, matrix-matrix multiplication, solving system of linear equations,

Sorting: Sorting networks, Bubble sort, Quick sort, Bucket sort and other sorting algorithms

Graph algorithms: Minimum spanning tree, single source shortest paths, all-pairs shortest paths, Transitive closure, connected components, algorithms for sparse graphs.

Learning Resources:

Text Books:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", Second Edition Pearson Education – 2007
2. Michael J. Quinn (2004), "Parallel Programming in C with MPI and OpenMP", McGraw-Hill International Editions, Computer Science Series,
3. Joseph Jaja, "An Introduction to Parallel Algorithms", Addison-Wesley, 1992

Reference Books:

1. S G Akl, The Design and Analysis of Parallel Algorithms, Prentice-Hall, 1989

CS16032 **3-0-0 (3)**

Human Computer Interaction

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the usability of interactive systems.
CO-2	Apply the techniques to manage the design process.
CO-3	Use the appropriate interaction style for a given problem.
CO-4	Design an interface for a given scenario based on the concepts of HCI.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	–	1	1	1
CO-2	2	1	–	2	2	2
CO-3	3	1	1	2	2	2
CO-4	3	2	2	3	3	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction: Usability of Interactive Systems, Universal Usability, Guidelines, Principles, and Theories

Design Processes: Design, Evaluation and the User Experience, Design Case Studies

Interaction Styles: Direct Manipulation and Immersive Environments, Fluid Navigation, Expressive Human and Command Languages, Devices

Design Issues: Advancing the User Experience, The Timely User Experience, Documentation and User Support, Information Search, Data Visualization.

Learning Resources:

Text Books:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos, Designing the User Interface: Strategies for Effective Human Computer Interaction, Pearson, 2021, 6th Edition

Reference Books:

1. Wilbert O Galitz, The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Wiley, 2007, 3rd Edition
2. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, Human-Computer Interaction, Pearson, 2004, 3rd Edition

CS16034 **3-0-0 (3)**

Information Retrieval

Pre-Requisites: **None**

Course Outcomes:

At the end of the course, the student will be able to

CO-1	Understand the basics of Information retrieval like what is a corpus, what is precision and recall of an IR system.
CO-2	Apply the data structures like Inverted Indices used in Information retrieval systems.
CO-3	Understand the basics of web search.
CO-4	Develop different techniques for compression of an index including the dictionary and its posting list.
CO-5	Analyze different components of an Information retrieval system.
CO-6	Develop the ability to develop a complete IR system from scratch.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	-	-	2	2	-	2
CO-2	2	-	2	3	-	2
CO-3	2	-	2	2	-	2
CO-4	2	-	2	3	-	2
CO-5	2	-	2	2	-	2
CO-6	2	-	2	2	-	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Boolean retrieval, the term vocabulary and postings lists, Dictionaries and tolerant retrieval, Introduction to index-construction and index-compression. Scoring, term weighting and the vector space model, Computing scores in a complete search system, Evaluation in information retrieval, Introduction to Relevance feedback and query expansion. Probabilistic information retrieval, review of basic probability theory, the probability ranking, principle, the binary independence model. Language models for information retrieval, Language modeling versus other approaches to IR, Text classification and Naive Bayes, Bayesian Network approaches to IR. Vector space classification, Support vector machines and machine learning on documents, Flat clustering, Hierarchical clustering, Matrix decomposition and latent semantic indexing. Introduction to Web search basics, Web crawling and indexes, Link analysis, Typical Assignments: Based on techniques studied, implementation of those techniques, study of research papers.

Learning Resources:

Text books

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, An Introduction to Information Retrieval, Cambridge University Press, Cambridge, England, 2009
2. Stefan Büttcher, Charles L, A. Clarke, Gordon V. Cormack, Information Retrieval: Implementing and evaluating search engines, MIT Press, 2010

Reference books

1. David A. Grossman, Ophir Frieder, Information Retrieval: Algorithms and Heuristics, Springer, 2004.
2. Frakes, Information Retrieval: Data Structures and Algorithms, Pearson, 2009.

CS16036 **3-0-0 (3)**

Internet of Things

Pre-Requisites: None

Course Outcomes:

CO-1	Understand current and future directions of Internet of Things.
CO-2	Design and develop communication protocols in Internet of Things.
CO-3	Develop smart environment and applications which advance the Internet of Things.
CO-4	Analyze the societal impact of Internet of Things.
CO-5	Analyze vulnerabilities, including recent attacks, involving the Internet of Things.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	-	-	2	2	1	-
CO-2	3	-	3	2	1	-
CO-3	2	-	2	3	1	3
CO-4	2	3	3	3	1	-
CO-5	2	-	3	3	1	-

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Internet of Things (IoT) frameworks and applications, IoT Standards, Smart Environments, Communication capabilities and Device Intelligence, Sensor and RFID Technology, Wireless Technologies for IoT, Zigbee/IEEE 802.15.4, IEEE 802.15.6 WBANs, Comparison of WPAN technologies, Mobile IPv6 for IoT, Machine-to-Machine communication models, Service Discovery in IoT, Service oriented Middleware, Resource management in IoT, Web of Things, Sensor Web, Crowd sourcing, Securing Internet of Things: vulnerabilities and attacks.

Learning Resources:

Text Books:

1. O Hersent, D Boswarthick and O Elloumi, The Internet of Things: Key applications and protocols, Wiley, 2012.
2. Daniel Minoli, Building The Internet Of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, John Wiley & Sons,2013.
3. Dieter Uckelmann, Mark Harrison and Florian Michahelles, Architecting the Internet of Things, Springer,2011.

Other Suggested Readings:

1. Nik Bessis and Ciprian Dobre, Big Data and Internet of Things: A Roadmap for Smart Environments, Springer,2014.
2. Giancarlo Fortino and Paolo Trunfio, Internet of Things Based on Smart Objects- Technology, Middleware and Applications, Springer,2014.

CS16038 **3-0-0 (3)**

Models in Deep Learning

Pre-Requisites: None

Course Outcomes:

CO-1	Identify Convolutional Neural Networks models to solve Supervised Learning Problems.
CO-2	Design Autoencoders to solve Unsupervised Learning problems.
CO-3	Apply Long Shot Term Memory (LSTM) Networks for time series analysis classification problems.
CO-4	Apply Classical Supervised Tasks for Image Denoising, Segmentation and Object detection problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	2	1	2	1	2
CO-2	2	1	2	2	3	1
CO-3	2	1	2	2	3	3
CO-4	2	2	3	2	3	3

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction to Biological Neurons, Artificial Neural Networks, McCulloch Pitts Neuron, Learning processes, Perceptron, Perceptron convergence theorem, XOR problem, Multilayer perceptron, Back Propagation (BP) Learning, Activation functions: Sigmoid, Linear, Tanh, ReLU, Leaky ReLU, SoftMax, loss functions, First and Second order optimization methods, Optimizers: Gradient Descent (GD), Batch Optimization, Momentum Based GD, Stochastic GD, AdaGrad, RMSProp, Adam; Introduction to Self Organizing Maps; Sequence to sequence models, RNN, Vanishing and Exploding Gradients, GRU, LSTM for NLP Applications; Convolutional Neural Network, Building blocks of CNN, Transfer Learning; Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Dropout; Autoencoders : Unsupervised Learning with Deep Network, Autoencoders, Stacked, Sparse, Denoising Autoencoders, Variational Autoencoders; Recent Trends in Deep Learning Architectures, Residual Network, Skip Connection Network, GoogleNet, DensenNet, SqueezNet, MobileNet, NasNet Models; Classical Supervised Tasks with Deep Learning, Segmentation Unet, FCN models, Object Localization (RCNN), FRCNN with Applications; Transformer, Generative Adversarial Network, Design own neural network models on Image, vision and NLP Applications.

Learning Resources:

Text Books:

1. Deep Learning- Ian Good felllow, Yoshua Benjio, Aaron Courville, The MIT Press.
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer,2006.
3. Simon Haykin, “Neural Networks, A Comprehensive Foundation”, 2nd Edition, Addison Wesley Longman, 2001.

CS16040 **3-0-0 (3)**

Models for Social Networks

Pre-Requisites: None

Course Outcomes:

CO-1	Develop random graph models for real-world networks.
CO-2	Understand the spread of information, disease, influence, etc., on networks.
CO-3	Design and develop models and algorithms for web search and sponsored search.
CO-4	Apply game-theoretic approaches to interaction on networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	2	2	-	2
CO-2	2	-	2	2	-	2
CO-3	2	-	2	2	-	2
CO-4	2	-	2	3	2	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Networks: Empirical Study of Networks, Technological networks, social networks, networks of information, biological networks.

Fundamentals of network theory: Mathematics for networks, Measures and metrics, the large-scale structure of networks; Overview of available network data: Newman's Graph data sets, SNAP Graph library.

Social network analysis software: Programming in Python and the Network X library, a distribution of Python for scientific computing and visualization.

Random Models of Networks: Random graphs: Basic properties of random graphs, degree distribution, clustering coefficient, giant component, small components; The Erdos - Renyi model of random graph; Inadequacy of the Erdos - Renyi model: A simple alternate random graph model; The Kleinberg result.

Random graphs with general degree distributions: The configuration model, Generating functions for degree distributions, Generating functions for the small components, Power-law degree distributions, Directed random graphs.

Models of network formation: The ``preferential attachment'' model of Barabasi and Albert, Vertex copying models, Stochastic Kronecker Graphs.

The Spread of "influence" through a Network: Stochastic Kronecker Graphs, The Christakis- Fowler work on the spread of obesity, happiness, etc. via social networks, modeling information cascades, Viral Marketing.

Spread of Disease on Networks: Random mixing models: SI, SIS, SIR, SIRS, basic differential equations, Basic Reproductive Number and analysis of branching processes, Analysis of SIR on the Configuration model, Synchronization in disease incidence, explanation via models, and observational studies from Syphilis, Example of studies with some specific diseases.

Information Networks: The structure of the web, Link analysis and web search, Page rank, Spectral Analysis of Page rank and hubs and authorities, random walks, Auctions and matching markets, Sponsored search markets.

Games on Networks: Basics of Game Theory: strategies, dominant strategies, dominated strategies, pure strategies and mixed strategies, Nash equilibrium, modeling network traffic as a game; Braess Paradox, Modeling Voluntary Vaccination as a game, Example of game- theoretic analysis applied to flu vaccine behavior.

Learning Resources:

Text Books:

1. Mark Newman, Networks: An Introduction, Oxford University Press,2010, First Edition.
2. David Easley and Jon Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press,2010, First Edition.

Other Suggested Readings:

1. https://archive.nptel.ac.in/courses/106/106/106106169/

https://ocw.mit.edu/courses/1-022-introduction-to-network-models-fall-2018/

CS16042 **3-0-0 (3)**

Natural Language Processing

Pre-Requisites: CS1603 Machine Learning

Course Outcomes:

CO-1	Understand the Text representation and Text pre-processing techniques.
CO-2	Understand language modeling with N-Grams.
CO-3	Apply syntactic parsing to produce parse trees
CO-4	Design NLP Systems for Text Summarization, Classification and Translation.
CO-5	Evaluate the performance NLP System.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	-	1	1	-
CO-2	1	-	1	1	2	-
CO-3	2	-	2	3	1	1
CO-4	2	-	2	4	3	2
CO-5	1	-	2	1	1	-

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction and Basic Text Processing, Text Representation- one-hot encoding, TF-IDF, Bag-of-Words, Word2Vec, Glove Embedding, Text-Preprocessing- Tokenization, Stemming-Porters Stemming algorithm, Lemmatization, Normalization, Spell Checker- Spelling Correction, Edit Distance, Language Modeling- Probabilistic Language Modelling- N-gram Modelling, Language Models Evaluation, Text Parsing, Part-of-Speech Tagging, POS with Hidden Markov model (HMM), Text Summarization-Abstractive and Extractive Text Summarization. Text Classification- Text Classification Techniques – Topic Modelling, Sentiment Analysis, Named Entity Recognition(NER), Machine Translation, Introduction to Large Language Models (LLMs)

Learning Resources:

Text Books:

1. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", 3rd Edition, 2008.
2. Uday Kamath, John Liu, James Whitaker, “Deep Learning for NLP and Speech Recognition”, Springer, 2020

Reference Books:

1. Allen, James, "Natural Language Understanding", Second Edition, Benjamin/ Cumming, 1995.
2. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, “Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems”, O’REILY, 2020

CS16044 **3-0-0 (3)**

Randomized and Approximation Algorithms

Pre-Requisites: None

Course Outcomes:

CO-1	Design and analyze efficient randomized algorithms.
CO-2	Apply tail inequalities to bound error-probability.
CO-3	Analyze randomized algorithms with respect to probability of error and expected running time.
CO-4	Analyze approximation algorithms and determine approximation factor.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	–	2	2	2	–
CO-2	3	–	2	2	2	–
CO-3	3	–	2	2	2	–
CO-4	3	–	2	2	2	–
CO-5	3	–	2	2	2	–

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction, Las Vegas and Monte Carlo Algorithms, Computational Model and Complexity Classes, Game Tree Evaluation, The Markov and Chebyshev Inequalities, The Stable Marriage Problem, The Coupon Collectors Problem, The Chernoff Bound, Routing in a Parallel Computer, The Probabilistic Method: Overview, probabilistic analysis, use of indicator random variables, Randomly permuting arrays, Birthday paradox, analysis using indicator random variables, Balls and bins, Streaks, Online hiring problem, Maximum Satisfiability, Expanding Graphs, The Lovasz Local Lemma, Markov Chains, Random Walks on Graphs, Graph Connectivity, Expanders and Rapidly Mixing Random Walks, Pattern Matching, Random Trees, Skip Lists, Hash Tables, Linear Programming, The Min-Cut Problem, Minimum Spanning Trees, The DNF Counting Problem, Maximal Independent Sets, Perfect Matching, The Online approximations paging Problem, Adversary Models and Paging against an Oblivious Adversary, Vertex cover problem, traveling salesman problem with triangle inequality, general traveling salesman problem, set-covering problem, a greedy approximation algorithm, analysis Randomization and linear programming, randomized approximation, subset-sum problem, Absolute approximations, Planar Graph Coloring, Maximum Programs Stored Problem, NP-hard Absolute Approximations-approximations, Polynomial time approximations schemes, Scheduling Independent Tasks , 0/1 Knapsack, Fully Polynomial time approximations scheme , Rounding , Interval Partitioning , Separation, probabilistically good algorithms.

Learning Resources:

Text Books:

1. Rajeev Motwani and Prabhakar Raghavan, Randomized Algorithms, Cambridge University Press,1995.
2. Vijay V. Vajirani, Approximation Algorithms, Springer, 2001, First Edition.

Reference Books:

1. J. Hromkovic, Design and Analysis of Randomized Algorithms, Springer, 2005.

Other Suggested Readings:

1. NPTEL Courses on Randomized and Approximation Algorithms

CS16046 **3-0-0 (3)**

Real Time Systems

Pre-Requisites: **None**

Course Outcomes:

CO-1	Understand the requirements of a real-time application and analyze the performance of different task scheduling algorithms for real-time systems.
CO-2	Understand the basic concepts of fault-tolerance and different fault-tolerance techniques available for real- time systems.
CO-3	Use simulated software to develop and test different fault tolerant models.
CO-4	Understand the concept of embedded systems and use various software tools for development of embedded systems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	1	2		1	
CO-2	3	1	2		1	
CO-3	2	1	2		1	
CO-4	3	1	2			

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to Real Time systems, applications of Real-Time systems, basic model of Real-Time systems, characteristics of Real-Time systems, types of Real-Time systems: hard, firm, soft, timing constraints, modelling timing constraints.

Real-Time task scheduling: basic concepts, clock driven scheduling, table driven scheduling, cyclic, schedulers, hybrid schedulers, event driven scheduling, EDF Scheduling, RMA, DMA, resource sharingamong RT tasks, Priority inversion, Priority Inheritance Protocol, Highest Locker Protocol, Priority Ceiling Protocol.

Introduction to Fault Tolerant Computing: Basic concepts and Fault tolerant scheduling of tasks Faults and their manifestations, Fault/error modelling, Reliability, availability and maintainability analysis, System evaluation, performance reliability trade-offs. System level fault diagnosis, Hardware and software redundancy techniques. Fault tolerant system design methods, Mobile computing and Mobile communication environment, Fault injection methods, Software fault tolerance, testing of fault tolerant software, fault modeling, built in self-test, data compression, error correcting codes, simulation of software/hardware, fault tolerant system design, CAD tools for design for testability.

Real-Time Embedded system, Need of well tested and debugged RTOS, Introduction to C/OS II. Case Studies of programming with RTOS: Smart card embedded system, Hardware and Software co-design: specification and design of an embedded system use of software tools for development of an embedded system. Recent advances in embedded applications.

Text Books

1. R. Mall, Real-Time Systems, Pearson, 2007
2. P. A. Laplante, Real-Time Systems Design & Analysis, Willey, 2011

Reference Books

1. S. V. Iyer & P. Gupat, Embedded Real-Time System Programming, Tata McGraw Hill, 2004

2. R. Kamal, Embedded System Architecture, Programming and Design, Tata McGraw Hill, 2007

CS16048 **3-0-0 (3)**

Reinforcement Learning

Pre-Requisites: nil

Course Outcomes:

CO-1	Understand elements of RL and solutions to Multi-Arm Bandit Problem.
CO-2	Formulate and solve problems modelled with Markov Reward Process.
CO-3	Comprehend solutions for problems with Markov Decision Process.
CO-4	Apply Dynamic Programming for Markov Decision Process
CO-5	Analyze Temporal Difference Methods

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	3	-	2	3	-	3
CO-2	3	-	2	3	-	3
CO-3	2	-	2	3	-	3
CO-4	2	-	2	3	-	3
CO-5	3	-	2	3	-	3

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction: Introduction to Reinforcement Learning (RL) – Difference between RL and Supervised Learning, RL and Unsupervised Learning. Elements of RL, Markov property, Markov chains, Markov reward process (MRP). Evaluative Feedback - Multi-Arm Bandit Problem: An n-Armed Bandit Problem, Exploration vs Exploitation principles, Action value methods, Incremental Implementation, tracking a non-stationary problem, optimistic initial values, upper-confidence-bound action selection, Gradient Bandits. Introduction to and proof of Bellman equations for MRPs. Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations. Dynamic Programming (DP): Overview of dynamic programming for MDP, principle of optimality, Policy Evaluation, Policy Improvement, policy iteration, value iteration, asynchronous DP, Generalized Policy Iteration. Monte Carlo Methods for Prediction and Control: Overview of Monte Carlo methods for model free RL, Monte Carlo Prediction, Monte Carlo estimation of action values, Monto Carlo Control, On policy and off policy learning, Importance sampling. Temporal Difference Methods: TD Prediction, Optimality of TD (0), TD Control methods - SARSA, QLearning and their variants.

Learning Resources:

Text Books:

1. Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, MIT Press 2020/Bradford Books 2018, Second Edition.

Reference Books:

1. Csaba Szepesvari, Algorithms for Reinforcement Learning, Morgan & Claypool, 2010, First edition.
2. Warren B. Powell, Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions, Wiley, 2022, First Edition.

CS16050 **3-0-0 (3)**

Security and Privacy

Pre-Requisites: nil

Course Outcomes:

CO-1	Evaluate the risks and vulnerabilities in protocols/Standards.
CO-2	Apply Number Theory and Algebra required for designing cryptographic algorithms.
CO-3	Design symmetric key and asymmetric key encryption techniques.
CO-4	Design authentication, message integrity and authenticated encryption protocols.
CO-5	Design and analyze security of systems including distributed storage and Electronic voting.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	-	3	3	-	-
CO-2	2	-	3	3	-	-
CO-3	2	-	3	3	-	-
CO-4	2	-	3	3	-	-
CO-5	2	-	3	3	-	-

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Security – risks, threats and vulnerabilities, Cryptography, Stream Ciphers – One-time Pad (OTP), Perfect secrecy, Pseudo-random generators (PRG), Attacks on stream ciphers and OTP, Real world stream ciphers, Semantic security, Case Study- RC4, Salsa 20, CSS in DVD encryption, A5 in GSM, Block ciphers- DES, attacks, AES, Block ciphers from PRG, Modes of operation – one-time key and many-time keys, CBC, CTR modes, Message Integrity – MAC, MAC based on PRF, NMAC, PMAC, Collision resistance – Birthday attack, Merkle-Damgard construction, HMC, Case study:SHA-256, Authenticated encryption, Key exchange algorithms, Public key cryptosystems – RSA, ElGamal, Elliptic curve cryptosystems – PKC, key exchange, IBE, Case studies – HTTPS – SSL/TLS, SSH, IPSec, 802.11i WPA, System design and analysis – Survivable distributed storage system, Electronic voting system.

Learning Resources:

Text Books:

1. J. Thomas Shaw, Information Security Privacy, ABA,2012.
2. J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC press,2008.
3. Menezes, et. al, Handbook of Applied Cryptography, CRC Press,2004.
4. A. Abraham, Computational Social Networks: Security and Privacy, Springer,2012.

CS16052 **3-0-0 (3)**

Service Oriented Architecture and Micro-Services

Pre-Requisites: nil

Course Outcomes:

CO-1	Understand software-oriented architectures.
CO-2	Design medium scale software project development using SOA principles.
CO-3	Develop SOA messages from business use cases.
CO-4	Design and implement modern SOA and SOA-specific methodologies, technologies and standards.
CO-5	Create composite services by applying composition style.
CO-6	Design Applications Using Microservices.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	-	1	1	-	-
CO-2	2	-	2	2	-	-
CO-3	2	-	2	2	-	1
CO-4	2	-	2	2	-	1
CO-5	2	-	2	2	-	1
CO-6	2	-	2	2	-	1

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Introduction To SOA, Evolution Of SOA: Fundamental SOA; Common Characteristics of contemporary SOA; Common tangible benefits of SOA; An SOA timeline (from XML to Web services to SOA); The continuing evolution of SOA (Standards organizations and Contributing vendors); The roots of SOA (comparing SOA to Past architectures). Web Services and Primitive SOA: The Web services framework· Services (as Webservices); Service descriptions (with WSDL); Messaging (with SOAP). Web Services And Contemporary SOA – I Message exchange patterns; Service activity; Coordination; Atomic Transactions; Business activities; Orchestration; Choreography. Web Services And Contemporary SOA-2: Addressing; Reliable messaging; Correlation; Polices; Metadata exchange; Security; Notification and eventing. Principles Of Service - Orientation: Services orientation and the enterprise; Anatomy of a service oriented architecture; Common Principles of Service orientation; How service orientation principles interrelate; Service orientation and object orientation; Native Web service support for service orientation principles. Service Layers: Service orientation and contemporary SOA; Service layer abstraction; Application service layer, Business service layer, Orchestration service layer; Agnostic services; Service layer configuration scenarios. Business Process Design: WS-BPEL language basics; WS Coordination overview; Service oriented business process design; WS addressing language basics; WS Reliable Messaging language basics. SOA Platforms: SOA platform basics; SOA support in J2EE; SOA support in. ET; Integration considerations.

Microservices: Introduction to Microservices, Challenges, SOA vs Microservices, Design and Implementation of Microservices.

Learning Resources:

Text Books:

1. Thomas Erl, Service-Oriented Architecture: Concepts, Technology and Design, Prentice Hall Publication, 2005.
2. Michael Rosen, Boris Lublinsky, Applied SOA Service Oriented Architecture and Design Strategies, Wiely India Edition, 2008.
3. Wolff, Eberhard. Microservices: flexible software architecture. Addison-Wesley Professional, 2016.

CS16054 **3-0-0 (3)**

SOCIAL MEDIA ANALYTICS

Pre-Requisites: **None**

Course Outcomes:

CO1	Classify social networks.
CO2	Analyze social media and networking data.
CO3	Apply Social networks Visualization tools.
CO4	Analyze the social data using graph theoretic computing approach.
CO5	Identify application driven virtual communities from social networks.
CO6	Apply sentiment mining.

Course Articulation Matrix:

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

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Introduction to social network and media analysis – Examples of Social Media and their characteristics, Society as a graph, candidates of social media and network data for analysis, Random graphs with general degree distributions, models of network formation, Properties of Large-Scale Networks: Six-degree separation, Scale-free distributions, Small-world effect, and strong community structure – strong and weak ties;

Social relatedness: networks and centrality Measures - degree, closeness, betweenness, edge betweenness, eccentricity, clustering coefficient, eigenvector; social media analytical applications.

Community Detection and graph-based clustering: communities in social media, node-centric community detection, group-centric community detection, network-centric community detection, hierarchy-centric community detection, Topology discovery, Community Evaluation;

Link Prediction: Challenges in link prediction, link prediction methods and algorithms, clustering approaches for link prediction;

Social Listening and Sentiment Analysis: Sentiments and Opinions, lexicon based methods, machine learning based methods, feature-based sentiment analysis, slang sentiment analysis;

Social Recommendation Systems: Classical recommendation algorithms – content-based methods, Collaborative Filtering, extending individual recommendation to groups of individuals; Recommendations using social context – using social context alone, extending classical methods with social context - Social Recommendation Using collaborative filtering, community detection and probabilistic matrix factorization, recommendations constrained by social context; evaluating recommendations.

Social Signal Processing: Understanding social interactions, social media content, speech and facial actions as social signals, Automatic analysis of social emotions, multimodal conversational analysis, SSP applications.

Information Diffusion in Social Media: Herd Behaviour – Bayesian Modeling of Herd Behaviour, Intervention; Information Cascades – Independent Cascade Model (ICM), Maximizing the spread of cascades, Intervention; Diffusion of Innovations – Innovation characteristics, diffusion of innovations models, modelling diffusion of Innovations, Intervention; Epidemics.

Learning Resources:

Text Books / Reference Books / Online Resources:

1. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, "*Social Media Mining – An Introduction*", Cambridge University Press, 2014.
2. Charu C Aggarwal (Ed.), "*Social Network Data Analytics*", Springer, 2011
3. Hansen, Derek, Ben Sheiderman, Marc Smith., "*Analyzing Social Media Networks with NodeXL: Insights from a Connected World*", Morgan Kaufmann, 2011.

CS16056 **3-0-0 (3)**

Soft Computing Techniques

Pre-Requisites: None

Course Outcomes:

CO-1	Understanding of optimization problems, comprehending the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.
CO-2	Understanding of the fundamental theory and concepts of neural networks, and Identifying different neural network architectures, algorithms, applications, and their limitations.
CO-3	Apply genetic algorithms and neural networks to solve real-world problems.
CO-4	Apply soft computing techniques to solve engineering and other societal problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5
CO-1	2	-	2	2	2
CO-2	2	2	2	2	-
CO-3	2	-	2	2	2
CO-4	2	-	2	2	2

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Overview of course and Basic of Soft Computing, Introduction of Neural Networks, Learning Process and Learning Task, Supervised Learning – Single and Multi – Layer Network, Associative Memory, Self-organizing Maps, Neuro-Dynamics, Hopfield Network, Fuzzy Logic and Systems-Fuzzy Sets and Membership Functions, Operations on Fuzzy Sets, Fuzzification. Fuzzy Numbers- Uncertain Fuzzy Values, Fuzzy Numbers and its L-R representation, Operations on Fuzzy Numbers. Fuzzy Relations, Fuzzy Inference Systems- Architecture of Fuzzy Inference System, Fuzzy Inference Rules and Reasoning, Defuzzification. Applications of Fuzzy Logic, Genetic algorithms and evolutionary computation. Applications of Genetic Algorithms & Hybrid Systems.

Learning Resources:

Text Books / Reference Books / Other Suggested Readings:

1. R. A. Aliev, R. R. Aliev, Soft Computing and Its Applications, World Scientific Publications, 2001.
2. Roger Jang, Tsai Sun, Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A computational Approach to Learning & Machine Intelligence, PHI, 2008.
3. Simon Haykin, Neural Network: A Comprehensive Foundation, PHI, 1999.
4. Kishan Mehtrotra, S. Ranka, Elements of artificial Neural Networks, Penram International Publishing (India), 2009
5. Timothy Ross, Fuzzy Logic with Engineering Applications, 3rd Edition, McGraw-Hill, 2010.
6. Bart Kosko, Neural Networks and Fuzzy Systems, PHI, 1994.

CS16058 **3-0-0 (3)**

Software Defined Networks

Pre-Requisites: Computer Networks

Course Outcomes:

CO-1	Understand basic knowledge of the use of SDN and its implementation.
CO-2	Design and implement a network topology using P4 and analyze various packet features.
CO-3	Ability to carry out further research in SDN and P4.
CO-4	Understand the Network Function Virtualization and its importance in Advanced communication.
CO-5	Apply the knowledge of networking to design future network problems.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	–	1	–	–	–	–
CO-2	1	–	–	–	–	–
CO-3	–	–	–	–	1	–
CO-4	–	–	2	–	–	–
CO-5	–	–	–	–	–	2

1 - Slightly; **2 - Moderately;** **3 - Substantially**

Syllabus:

Network layer – Life of a Packet, Introduction and What's inside a router, Networks without challenges facing network managers, How SDN Works and SDN Architecture, SDN Controllers OpenFlow Overview-OVS, OpenFlow-Flow Tables and SDN use cases, Mininet, Existing SDN Controllers- Floodlight and Open Daylight, SDN Controller-Link discovery, Topology management, Flow manager, Decision making, Controller placement problem, Multi-controller issues, Load Balancing, Network measurement, network verification, network security, and network traffic management, Traffic engineering, Network Function Virtualization, Smart NIC, Data Plane-Motivation, Data Plane Implementation and Introduction to P4, Introduction to P4 & PISA architecture, P4 language, Software-Defined Networking for Internet of Things, Security issues on SDN architecture.

Learning Resources:

Text Books:

1. P. Goransson, C. Black, and T. Culver, Software Defined Networks: A Comprehensive Approach, Morgan Kaufmann, 2017, 2nd Edition.
2. K. Gray and T. D. Nadeau, Network Function Virtualization. Morgan Kaufmann, 2016.
3. Kreutz, D., Ramos, F. M. V., Veríssimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. , Software-defined networking: a comprehensive survey. CoRR., 2018.

Reference Books:

1. Qiang Duan Mehmet Toy, Virtualized Software-Defined Networks And Services, Artech House, 2016. First Edition
2. Siamak Azodolmolky and Oswald, CokerSoftware-Defined Networking with OpenFlow, Packt Publishing, 2017, Second Edition

CS16060 **3-0-0 (3)**

Vehicular Networks

Pre-Requisites: None

Course Outcomes: At the end of the course the student will be able to:

CO-1	Understand the knowledge of Wireless Vehicular AdHoc Networks.
CO-2	Understand the knowledge of Sensor Technology.
CO-3	Analyze the Routing Protocols of Vehicular AdHoc Networks.
CO-4	Understand the Data Dissemination in Vehicular Networks.

Course Articulation Matrix:

	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	1	1	1	1	1	1
CO-2	1	1	1	-	1	1
CO-3	-	1	1	1	1	1
CO-4	-	1	1	1	1	1

1 - Slightly; 2 - Moderately; 3 - Substantially

Syllabus:

Introduction to Vehicular AdHoc Networks (VANETs), Traffic Monitoring, Causes of Congestion, Traffic Monitoring Data, Commonly Used Sensor Technology, Vehicle Movements Model, Vehicle-to-Vehicle and Infrastructure-to-Vehicle Communication, Routing in MANETs, Routing Protocols for VANETs, Delay Tolerant Networks in VANETs, Vehicle Traffic Model, Vehicle-Roadside Data Access, Data Dissemination in VANETs, Localizations in VANETs, Localization aware VANETs Applications, Localization Techniques for VANETs, Data Fusion in VANET Localization System.

Learning Resources:

Text Books/ Reference Books:

1. Stephan Olariu, Michele C. (Eds), Vehicular Networks: From theory to practice, Chapman and Hall/CRC, 2009,1.
2. Claudia Campolo, Antonella Molinaro, Riccardo Scopigno (Eds), Vehicular ad hoc Networks., Springer Cham, 2015, 1.