DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M. TECH. COMPUTER SCIENCE & ENGINEERING

TWO-YEAR PROGRAMME

CURRICULUM STRUCTURE & SYLLABI
(2023 – 25 ONWARDS)

SCHOOL OF ENGINEERING AND TECHNOLOGY
MODY UNIVERSITY OF SCIENCE AND TECHNOLOGY
LAKSHMANGARH

M. Tech. Computer Science & Engineering

Two-Year Programme

Academic Curriculum (2023 – 25 onwards)

Curriculum Component

Component	Credit
1. University core	
2. University Elective	
3. Basic Science (3 Courses)	7
4. Engineering Science	
5. Program Core (12 Courses)	33
6. Program Elective (2 Courses)	8
7.Project/Summer Internship Project (4 Courses)	32
8. SEFS (Non Graded)	
Total	**80**

Credit Distribution across all Components									
Semester	**UC**	**UE**	**BS**	**ES**	**PC**	**PE**	**SIP**	**Proj**	**Total**
First		-	7	-	14	4	-	0	25
Second	-		-	-	19	4	-	0	23
Third	-	-	-	-	-	-	-	16	16
Fourth	-	-	-	-	-	-	-	16	16
Total	-		7		33	8		32	80

M. Tech. Computer Science and Engineering

Two-Year Programme

Academic Curriculum (2023 – 25 onwards)

First Year

	Course Code	Course Title	Contact Hours per Week			Credits	ETE Duratio	Weightage (%)		
			L	T	P		Hours	CW	MTE	ETE
Autumn Semester	CS 19. 521	Big Data Analytics	3	1	0	4	3	10	40	50
	CS 19.511	Adhoc Sensor Networks	3	1	0	4	3	10	40	50
	CS 19.515	Machine Learning	3	1	0	4	3	10	40	50
	CS 19. 523	Operations Research	3	0	0	3	3	10	40	50
		Program Elective I	4	0	0	4	3	10	40	50
	RM 19.101	Research Methodology	3	0	0	3	3	10	40	50
	CS 19.517	Adhoc Sensor Networks Lab	0	0	2	1	2	20	40	40
	CS 19. 525	Machine Learning Lab	0	0	2	1	2	20	40	40
	CS 19. 527	Operations Research Lab	0	0	2	1	2	20	40	40
		SEFS (Non Graded)				1				
	Sub		19	3	6	25				

	Course Code	Course Title	Contact Hours per Week			Credits	ETE Duratio	Weightage (%)			
			L	T	P		Hours	CW	MTE	ETE	
Spring Semester	CS19.506	Distributed Operating Systems	3	1	0	4	3	10	40	50	
	CS19.508	Soft computing Paradigms	3	1	0	4	3	10	40	50	
	CS19.510	Digital Image Processing & Analysis	3	1	0	4	3	10	40	50	
	CS19.512	Advances in Database Management Systems	3	1	0	4	3	10	40	50	
	PE	Program Elective II	4	0	0	4	3	10	40	50	
	CS19.514	Distributed Operating Systems lab	0	0	2	1	2	20	40	40	
	CS19.516	Digital Image Processing & Analysis lab	0	0	2	1	2	20	40	40	
	CS19.518	ADBMS lab	0	0	2	1	2	20	40	40	
		SEFS (Non Graded)				1					
	Sub Total		16	4	6	23					

List of Electives

Program Elective I

1. CS 19.563 Advances in Distributed Computing
2. CS 19.565 Advanced Compiler Construction
3. CS 19.567 BioInformatics
4. CS 19.569 Advanced Computer Graphics
5. CS 19.571 Advances in Algorithm Design

Program Elective II

1. CS 19.562 Mobile Cloud Computing
2. CS 19.564 Machine Intelligence & Robotics
3. CS 19.566 Natural Language Processing
4. CS 19.568 Advanced Software Engineering
5. CS 19.570 Mathematical Modelling

M. Tech. Computer Science & Engineering

Two-Year Programme

Academic Curriculum (2023 – 25 onwards)

Second Year

	Course Code	Course Title	Contact Hours per			Credits	ETE Duration	Weightage (%)		
			L	T	P		Hours	CW	MTE	ETE
Autumn Semester	CS 19.601	Seminar	0	0	--	4	-	100		
	CS 19.603	Project	0	0	--	4	-	100		
	CS 19.608	Dissertation Preliminary	0	0	--	8	-	100		
	Sub total		**2**	**0**	**--**	**16**				

	Course Code	Course Title	Contact Hours per Week			Credits	ETE Duration	Weightage (%)		
			L	T	P		Hours	CW	MTE	ETE
Spring Semester	CS 19.606	Dissertation Final	0	0	--	16	-	100		
	Sub Total		**0**	**0**	**--**	**16**				

Total Credits = 80

M. Tech. Computer Science & Engineering

Two-Year Programme
Academic Curriculum (2023 – 25 onwards)

CS 19.521 **Big Data Analytics** **Total Lectures: 40**

Credit:3-1-0-4

Prerequisite: Data Mining, Database Management Systems

Objective(s):

- To learn Big Data characteristics
- To learn Hadoop architecture and functioning of Map-Reduce.
- To learn other Big data tools like HIVE, HIVEQL and HBASE.

1. **INTRODUCTION: D**istributed file system – Big Data and its importance, four Vs, drivers for Big data, Big data analytics, Big data applications, algorithms using map reduce, matrix-vector multiplication by Map Reduce. [8]

2. **Introduction to HADOOP:** Apache HADOOP & HADOOP ecosystem , moving data in and out of Hadoop , understanding inputs and outputs of MapReduce , data serialization [8]

3. **HADOOP Architecture:** HADOOP architecture, HADOOP storage: HDFS, common HADOOP shell commands , anatomy of file write and read., namenode, secondary namenode, and datanode, HADOOP MapReduce paradigm, map and reduce tasks, job, task trackers ,cluster setup, SSH & HADOOP configuration, HDFS administering, monitoring & maintenance. [8]

4. **HADOOP Ecosystem and YARN**: HADOOP ecosystem components, schedulers, fair and capacity, HADOOP 2.0 new features, namenode, high availability, HDFS federation, mrv2, YARN, running mrv1 in YARN. [8]

5. **HIVE and HIVEQL, HBASE**: HIVE architecture and installation, comparison with traditional database, hiveql, querying data, sorting and aggregating, map reduce scripts, joins & subqueries, Hbase concepts, advanced usage, schema design, advance indexing - Pig, Zookeeper - how it helps in monitoring a cluster, Hbase uses Zookeeper and how to build applications with Zookeeper. [8]

Outcome(s): The students will be able to carry out research in Big Data using the latest tools.

Text Books:

[1] Chris Eaton, Dirk deroos et al., Understanding Big data , McGraw Hill, 2012.

[2] Tom White, HADOOP: The definitive Guide , O Reilly 2012.

Reference Books:

[1] Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, Professional Hadoop Solutions, Wiley, 2015.

[2] Jy Liebowitz, Big Data and Business analytics, CRC press, 2013.

CS 19.511 **Ad hoc Sensor Networks** **Total Lectures: 40**

Prerequisite: Computer Networks **Credit: 3-1-0-4**

Objective(s):

- To learn the Ad hoc networks and related protocols
- To learn QoS parameters of Ad hoc network
- To learn the basic principles of sensor networks
- To understand the functionality of wireless sensor network and respective securities issues

1	**Ad hoc Networks:** Introduction and Definitions, Ad hoc Network Applications, Design Challenges. Routing in Mobile Ad hoc Networks: Introduction, Flooding, Proactive Routing, On Demand Routing, Proactive Versus on demand Debate, Location based Routing.	[7]
2	**Multicasting and Transport layer Protocols in Ad hoc Networks:** Introduction, Classifications of Protocols, Multicasting Protocols, Broadcasting. Protocol Comparisons, Overarching Issues. Transport layer Protocols in Ad hoc Networks: Introduction, TCP and Ad hoc Networks, Transport Layer for Ad hoc Networks: Overview, Modified TCP, TCP-aware Cross-layered Solutions. Ad hoc Transport Protocol.	[9]
3	**QoS Issue in Ad hoc Networks:** Introduction, Definition of QoS, Medium Access Layer, QoS Routing, Inter- Layer Design Approaches. Security in Mobile Ad hoc Networks: Vulnerabilities of Mobile Ad hoc Networks, Potential Attacks, Attack Prevention Techniques. Intrusion Detection Techniques.	[8]
4	**Overview of Wireless Sensor Networks (WSNs)**: Introduction, Constraints and Challenges, Advantages of Sensor Networks, Sensor Network Applications' Routing Protocols: Geographic, Energy-Aware Routing – Unicast Geographic Routing, Routing on a Curve, Energy-Minimizing Broadcast, Energy-Aware Routing to a Region; Attribute-Based Routing – Directed Diffusion, Rumor Routing.	[10]
5	**Security in WSNs:** Introduction: IEEE 802.11b Security Mechanisms, WEP Security Issues, WEP Improvements, Wireless Security Threats and Risks: Types of Security Threats, Physical Security of Wireless Devices, WLAN Attacks. Risk Mitigation and Countermeasures: Basic Countermeasures, AAA Infrastructure Solutions, Additional Enhancements.	[6]

Outcome(s): On successful completion of this course, students will be able to
Understand the basic working of sensor networks
Understand security aspects of WSN

Text Books:

[1] Prasant Mohapatra and Srihanamurthy, "Ad Hoc Networks Technologies and Protocols", Springer, Springer International Edition, 2009.

[2] Kazem Sohraby, Daniel Minoli, Taieb Znati, "Wireless Sensor Networks", John Wiley & Sons, Inc., Pub!ication, 2007.

Reference Books:

[1] A.Boukereche, "Handbook of Algorithms for Wireless Networking and Mobile Computing", Chapman and Hall/CRC Computer and Information Science Series, 2007.

[2] Zhao, Guibas, "Wireless Sensor Networks" , Elsevier, 2004.

CS 19.515 **Machine Learning** **Total Lectures: 40**

Credit:3-1-0-4

Prerequisite: Artificial Intelligence

Objective(s):

- To learn the Machine learning concepts.
- To learn different models of Machine learning.

1. **Foundations of Learning:** Components of learning, learning models, geometric models, probabilistic models, logic models, grouping and grading, learning versus design, supervised learning, unsupervised learning, reinforcement learning , theory of learning, feasibility of learning, error and noise, training versus testing, theory of generalization, generalization bound, approximation generalization trade off, bias and variance, learning curve. [8]

2. **Linear Models:** Linear classification, univariate linear regression, multivariate linear regression, regularized regression, logistic regression, perceptron, multilayer neural networks, learning neural networks structures, support vector machines, soft margin SVM, going beyond linearity, generalization and overfitting, regularization, validation. [8]

3. **Distance-based Models:** Nearest neighbour models, K-means clustering, hierarchical clustering, k-d trees, locality sensitive hashing, non-parametric regression, ensemble learning, bagging and random forests, boosting, meta learning. [8]

4. **Tree and Rule Models:** Decision trees, learning decision trees, ranking and probability estimation trees, regression trees, clustering trees, learning ordered rule lists, learning unordered rule lists, descriptive rule learning, association rule mining, first-order rule learning. [8]

5. **Reinforcement Learning:** Passive reinforcement learning, direct utility estimation, adaptive dynamic programming, temporal-difference learning, active reinforcement learning, exploration, learning an action utility function, generalization in reinforcement learning, policy search, applications in game playing, applications in robot control. . [8]

Outcome(s): Students will be able to

- Compare and apply Machine learning models for different problems.
- Carry out research in the area of Machine learning.

Text Books:

[1] T. M. Mitchell, Machine Learning, McGraw Hill, 1997.
[2] P. Flach, Machine Learning: The art and science of algorithms that make sense of data, Cambridge University Press, 2012.

Reference Books:

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2012.

[2] S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall, 2009.

CS 19.523 **Operations Research** **Total Lectures: 40**
Credit: 3-0-0-3

Prerequisite: Engineering Mathematics
Objective(s): To learn major Operations research techniques.

1. **Introduction to Operation Research:** Origins of or, nature, impact and phases of OR, solving the OR Model , operation research as tool for decision support system, productivity improvement, overview of OR research techniques. [6]

2. **Linear Programming Extensions:** Formulation of linear programming problem, linear programming models, assumptions of linear programming, graphical method of solving LP problem, Simplex method for solving LP problem, special cases in Simplex method application. Formulation of transportation problem, types of transportation problems, methods of initial feasible solution, methods of optimum solution, unbalanced transportation problem, assignment problem, transshipment model. [14]

3. **Decision, Game & Queueing Theory:** Formulation of two person, zero-sum games, solving simple games, mixed strategies, non-zero sum games, basic structure & components of decision, decision criteria, decision trees, basic characteristics of queueing system, terminologies & notation, Poisson process of queueing, M/M/1 system queueing model. queueing decision models. [8]

4. **Hybrid OR Models, Project Management PERT & CPM:** Assumption and comparison PERT & CPM, algorithms of PERT CPM techniques, fundamentals of network model, guidelines for network construction, critical path analysis, methods based on time estimates to find critical paths, concept of slack & oats in network analysis, Project Evaluation & Review Techniques (PERT) [6]

5. **Dynamic Programming:** Terminologies, multi decision process, Bellman's principles of optimality, characteristics of dynamic programming problems, dynamic programming algorithms, solving LPP using dynamic programming, recent applications of dynamic programming in OR. [6]

Outcome(s): Students will be able to apply Operation Research techniques relevant to their area of research.

Text Books:
[1] Hamdy A Taha, Operations Research: An Introduction, Pearson education India, 8th Edition, 2011.
[2] Frederick S. Hillier, Gerald J. Lieberman, Introduction to Operations Research, 8th edition McGraw-Hill, 2001.

Reference Books:
[1] Richard Bronson , Govindasami Naadimuthu, Schaum's Outline of Operations Research, 2nd edition, McGraw-Hill Education, 1997.
[2] Wayne L. Winston, Operations Research: Applications and Algorithms, Cengage Learning, 2003.

RM 19.101 **Research Methodology** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite(s): Communication Skills

Objective(s):

- To understand a general definition of research design.
- To identify the overall process of designing a research study from its inception to its report.
- To be familiar with ethical issues in educational research, including those issues that arise in using quantitative and qualitative research.
- To understand the primary characteristics of quantitative research and qualitative research.
- To identify a research problem stated in a study

1. **Introduction**: Meaning of research, Objectives, Motivation for Research, Types of Research, Research Approaches, Research Process, Validity and Reliability in Research. [6]
2. **Problem Formulation**: Identification, Selection and Formulation of a Research Problem, Criteria of a good Research Problem, Review of Literature, Research Gaps. [10]
3. **Research Design**: Research Framework, Meaning &Significance of Research Designs, Features of a good Research Design, Types of Research Design. [10]
4. **Data Analysis and Simulation Tool**: Types of Data, Sources of Data, Methods of Collecting Data, Data Presentation Techniques. Methods of Data Analysis, SPSS, MATLAB, Network Simulators, ETAP, Solid Works, MultiSim. [12]
5. **Report Writing**: Types of Reports, Contents, Style Manuals, Results & Findings, Contributions, Implications, Scope for future work and conclusion, Referencing Styles, Anti Plagiarism Policy. [2]

Outcome(s): On successful completion of this course, the students will be able to

- Assess critically the following methods: literature study, case study, structured surveys, interviews, focus groups, participatory approaches, narrative analysis, cost-benefit analysis, scenario methodology and technology foresight.

Text Books:

1. R.Pannershelvam, "ResearchMethodology" PrenticeHall India, New Delhi, 2nd Edition, 2013.
2. C.R.Kothari,"ResearchMethodology: MethodsandTechniques",New Age International (p) Limited,Publishers New Delhi, 4ndedition,2018.

Reference Books:

1 William G.Zikmund, “Business Research Methods”, Cengage Learning New Delhi, 8th edition, 2016.
2. Mark Saunders, “Research Methods for Business Students” Pearson Education New Delhi, 7th edition, 2015.
3. Donald R Cooper, “Business Research Methods” Tata McGraw-Hill Publishing Company Limited New Delhi, 12th edition, 2013

CS 19.517 Ad-hoc Sensor Networks Laboratory Total Hours: 26

Credit: 0-0-2-1

Prerequisite: Computer Network

Objectives: The objective of this course is to:

- Become comfortable with network simulation
- Provide the foundation of to explore simulation of various types networks.

Contents:

- Basics of Network Simulation
- Simulating a Local Area Network
- Measuring Network Performance
- Simulation of a Satellite Network
- Simulating a Wi-Fi Network
- Simulating a Wi-MAX Network
- Simulating a Mobile Ad-hoc Network
- Simulating a Wireless Sensor Network
- Setting up a Bluetooth Network
- Setting up a Zig-Bee Network

Outcomes: Upon completion of this course, students will be:

- Able to simulate different types of networks and performance measures of network
- Able to setting up Bluetooth and Zig-Bee network

CS 19.525 **Machine Learning Lab** **Total Hours: 26**

Credit: 0-0-2-1

Prerequisite: Artificial Intelligence

Objective(s): To learn the implementation of different Machine learning models.

Contents:

Experiments based upon basic learning models, classification, generalization.

Experiments based upon linear models of learning.

Experiments based upon distance models.

Experiments based upon tree and rule models.

Experiments based upon reinforcement learning.

Outcome(s): Students will be able to apply and compare different Machine learning models.

CS 19.527 **Operations Research Lab** **Total Hours: 26**

Credit: 0-0-2-1

Prerequisite: Engineering Mathematics

Objective(s): To implement the Operations Research techniques.

Contents:

Experiments based upon linear programming.

Experiments on Simplex method.

Experiments based upon Assignment problem and Transportation model.

Experiments based upon Game theory and Queuing theory.

Experiments based upon CPM/PERT.

Experiments based upon dynamic programming.

Outcome(s): Students will be able to implement and analyze different OR techniques.

CS 19.506 **Distributed Operating Systems** **Total Lectures: 40**

Credit: 3-0-0-3

Prerequisite: Operating System

Objective(s):

- To learn the operating system and distributed computing environment
- To learn remote procedure call and Remote method invocation.
- To learn the principles of distributed shared memory
- To learn distributed file system..

1	**Introduction and distributed computing environment**: Definition, Evolution of distributed computing system, Distributed computing system models- Minicomputer model, workstation model, workstation –server model, processor –pool model, hybrid model. Advantages of distributed operating system. Design issues of distributed operating system- Transparency, Reliability, concurrency, scalability, heterogeneity, openness, security.	[8]
2	**Message Passing:** Introduction, issues in IPC by message passing, Synchronization, buffering –null buffer, single message buffer, unbounded capacity buffer, finite bound buffer. Multi datagram message, encoding and decoding of message data. Process addressing, failure handling, group communication.	[6]
3	**Remote Procedure call and Remote Method Invocation:** RPC model, Transparency of RPC, Implementing RPC mechanism, Stub generation, RPC Messages, Parameter passing, semantics, call semantics. Communication protocols for RPC-request protocol, request/reply protocol, request/reply/acknowledge protocol, RPC heterogeneous environment, Light weight RPC. Remote Method Invocation(RMI) : Introduction, communication between distributed objects, design issues for RMI implementation of RMI , clock synchronization, logical clock- Lamport time stamps, deadlock, election algorithms- Bully algorithm, Ring algorithm , distributed transactions.	[15]
4	**Distributed shared memory:** Introduction, message passing vs DSM, implementation approaches to DSM, design and implementation issues- Structure and synchronization model. Granularity, Structure of shared memory space, consistency model, replacement strategy, thrashing.	[8]
5	**Distributed File System:** Design, implementation, trends in distributed file system.	[7]

Outcome(s): On successful completion of this course, students will be able to

- Understand the primary functionalities of operating system.
- Understand Message passing efficiently in an operating system environment
- Develop new operating systems as well as work on existing ones to develop them further

Text Books:

[1] Pradeep K. Sinha, “Distributed Operating Systems: Concepts and Design”, Prentice-Hall of India, 3rd Printing, October 2001 .
[2] Andrew S. Tanenbaum and Maarten van Steen, “ Distributed Systems”, PHI, second Printing, July 2003.

Reference Books:

[1] G. Coulorius, J. Dollimore and T. Kinderberg, “Distributed Systems: Concepts and Design”, fourth edition, Addison Wesley, May 2005.

CS 19.508 **Soft Computing Paradigms** **Total Lectures: 40**

Credit: 3-1-0-4

Prerequisite: Artificial Intelligence

Objective(s):

- To familiarize with soft computing concepts
- To introduce the idea of artificial neural networks, fuzzy logic and use of heuristics based on human experience
- To introduce the concept of genetic algorithm and its applications to soft computing using some applications

1. **Artificial Neural Networks:** Definition, benefits of Artificial Neural Networks, terminology, neuron models, activation function, network architectures, learning process, types of learning: Hebbian, competitive, error-correction, Boltzmann learning. [8]

2. **Types of Neural Networks:** Feed forward neural network: Single layer perceptron, limitations, multi layer perceptron, back propagation algorithm, practical considerations, radial basis function network. Recurrent networks: Hopfield network, NARX model, state space model, recurrent multi layer, perceptron, second order networks. Self-organizing map, and principal component analysis. [12]

3. **Fuzzy Logic and applications :** Basic concepts of fuzzy logic, crisp sets and fuzzy sets, operations and properties of crisp sets and fuzzy sets, crisp and fuzzy relations, fuzzy rules, fuzzification and defuzzification. Pattern recognition, control engineering, image processing, neuro fuzzy application. [10]

4. **Genetic Algorithms:** Basic concepts, working principles, genetic operators, genetic programming, parsing trees and mathematical foundation of genetic algorithm. . Other optimization methods: Swarm Intelligence, ant colony optimization. [6]

5. **Hybrid Systems:** Neuro genetic systems, fuzzy genetic systems and neuro fuzzy genetic systems, applications. [4]

Outcome(s): On successful completion the students will be able to:

- Identify and describe soft computing techniques and their roles in building intelligent machines
- Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
- Apply genetic algorithms to combinatorial optimization problems
- Apply neural networks to pattern classification and regression problems
- Evaluate and compare solutions by various soft computing approaches for a given problem

Text Books:

[1] Li Min Fu, “Neural Networks in Computer Intelligence”, Tata McGraw Hill Edition, 2008.

[2] S.N. Sivanandam, S. N. Deepa, “Principles of Soft Computing”, Wiley India, 2008.

Reference Books:

[1] S. Rajsekaran, G. A. Vijayalaxmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Application”, Prentice Hall, 2010.

[2] Simon S. Haykin , “Neural Networks- A Comprehensive Foundation”, Prentice Hall, 2009

[3] D.E. GoldBerg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Pearson Education, 2009.

[4] S. Rajsekaran, G. A. Vijayalaxmi Pai, “Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Application”, Prentice Hall, 2010

CS 19.510 **Digital Image Processing and Analysis** **Total Lectures: 40**

Credit: 3-1-0-4

Prerequisite: Computer Programming, Computer Graphics

Objective(s):

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To study the image enhancement techniques
- To study image restoration procedures.
- To study the image compression procedures.
- To study the image segmentation and representation techniques
- To study the image representation in form of features and descriptors

1. **Image Representation:** Sampling, Quantization, Image Basis Function, Image Types, Fourier Transform, Discrete Cosine Transform, Walsh- Hadamard Transform, Wavelet Transform, Principal Component Analysis [6]

2. **Image Enhancement and Restoration:** Contrast Stretching, Histogram Specification, Histogram Equalization, Enhancement Using Arithmetic/Logic Operations, Smoothing Sharpening, Low-Pass Filters, High-Pass Filters, Bandpass Filters, Homomorphic Filtering, Restoration Process Model, Noise Models, Inverse Filtering , Minimum Mean-Square Error Restoration. [8]

3. **Image Morphology**: Introduction, Logic Operations Involving Binary Images, Dilation and Erosion, Opening And Closing, Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening. [5]

4. **Image Segmentation:** Region Extraction, Pixel-Based Approach, Thresholding, Edge And Line Detection, Pattern Fitting Approach, Hough Transform, Radon Transform, Edge Linking And Edge Following, Edge, Corner Detection**.** [6]

5. **Pattern Recognition , Compression and Feature Extraction** [15]
Recognition: Deterministic Methods, Clustering, Neural Network, Template Matching, Statistical Classification, Syntactic Recognition, Tree Search, Graph Matching.Image Compression: Information Content of an Image, Lossless and Lossy Compression Algorithms, Compression Standards. Feature Extraction: Representation, Topological Attributes, Geometric Attributes, spatial Moments, Texture, Boundary Based Description, Region Based Description, Intensity Based and Relational Description.

Outcome(s): On successful completion of this course, students will be able to

- Discuss digital image fundamentals.
- Apply image enhancement and restoration techniques.
- Use image compression, segmentation and features extraction Techniques.

Text Books:

[1] Gonzaloz R. C. and Woods R.E., “Digital Image Processing”, Prentice Hall 2002.

[2] Jain A.K., “Fundamentals of Digital Image Processing”, Prentice Hall 1989.

Reference Books:

[1] William K. Pratt, “Digital Image Processing”, John Wiley, 2001.
[2] B. Chanda and D. Dutta Majumder ,” Digital Image Processing and Analysis”,PHI, 2002

CS 19.512 **Advances in Database Management Systems** **Total Lectures: 40**

Credit: 3-1-0-4

Prerequisite: Database Management System

Objective(s): The objectives of this Course are :

- To learn object relational databases
- To learn query processing and optimization
- To learn database recovery and database security
- To learn distributed databases

1. **Object and Object-Relational Databases:** Introduction, Object relational features, Object database extensions to SQL, ODMG object model and the object definition language (ODL) [8]

2. **Query Processing and Optimization:** Translating SQL queries into relational algebra. Algorithms for external sorting, select, join, project and set operations. Implementing aggregate operations and outer joins. Combining operations using pipelining, heuristic query optimization, selectivity and cost estimates in query optimization, semantic query optimization. [8]

3. **Database Recovery:** Recovery concepts, no-undo/redo recovery based on deferred update, recovery techniques based on immediate update, shadow paging, ARIES algorithm, recovery in multi database systems, database backup and recovery from catastrophic failures. [8]

4. **Database Security:** Security issues, granting and revoking privileges, role based access control for multilevel security, SQL injection, statistical database security, flow control, encryption and public key infrastructures, privacy issues and preservation, challenges of database security**.** [8]

5. **Distributed Databases:** Distributed database architecture, fragmentation, and query processing and optimization management in distributed databases. Database concepts, types, distributed distribution and allocation techniques, in distributed databases, transaction [8]

Outcome(s): After completion of the course, students will be able to:

- Understand object and object relational databases.
- Understand Structured Query Language (SQL) with query optimization.
- Understand issues in database recovery and database security

Text Books:

[1] Elmasri, Navathe, "Fundamentals of Database systems", Addison Wesley, sixth ed, 2010.

[2] Raghu Ramakrishnan, Gehrke, "Database Management systems", McGraw Hill, third ed, 2009.

Reference Books:

[1] Thomas m. Connolly, Carolyn E. Begg, "Database Systems", Pearson education, fifth ed, 2009.

CS 19.514 **Distributed Operating Systems Laboratory** **Total Hours: 26**

Credit: 0-0-2-1

Prerequisite: Operating System Laboratory

Objectives: The objective of this course is to:

- To develop skills to design and analyze Inter process communication in shared memory
- To Strengthen the ability to implement Operating system in distributed environment

Contents:

- Study of UNIX commands with all their important options
- Study system calls related to process & process control
- Study system calls related to semaphore
- Inter process communication using shared memory
- Incrementing a counter in shared memory
- Implement concurrent echo client-server application
- Implement a distributed chat server using TCP sockets in java.
- Simulation of Distributed mutual exclusion in java.
- Write a program to implement RPC in "C"
- Implement „Java RMI" mechanism for accessing methods of remote systems.
- Write a program to simulate the functionality of Lamport's Logical clock in C.
- Write a program to Implement Vector clock in C.

Objectives:

Upon completion of this course, students will be able to :

- Understand process and process control and semaphore
- Understand working of client server application
- Understand working of TCP sockets in Java

CS 19.516 Digital Image Processing and Analysis Laboratory Total Hours: 26

Credit: 0-0-2-1

Prerequisite: Computer Programming Laboratory

Objectives:

The objective of this course is to:

- To learn MATLAB environment for image processing
- To learn various types of functions Image restoration, Region extraction and features extraction.

Contents:

- Overview of MATLAB environment, Programming, Data types, Structure, Arithmetic Logical operations, Cell arrays, Script files, Functions.
- Implement following image processing concepts using MATLAB.
- Fourier transform, discrete cosine transform, Walsh- Hadamard transforms and principal component analysis.
- Contrast Stretching, Histogram Equalization, and Enhancement using Arithmetic/Logic Operations.
- Smoothing, Mean filter, Ordered Statistic Filter, Sharpening, Low-pass, High-pass filters, Bandpass and Homomorphic Filtering.
- Image Restoration using Inverse filtering, Minimum Mean-square Error Restoration.
- Logic Operations involving Binary Images, Dilation and Erosion, Opening and Closing, Boundary Extraction, Region Filing, Extraction of Connected Components, Convex Hull, Thinning, Thickening.
- Region Extraction, Thresholding, Edge and Line Detection using gradient operators, Hough transform, Radon transform.
- Object representation, topological attributes, Extraction of object by Boundary based descriptors.
- Feature Extraction of object by Region based descriptors.
- Object recognition problem with preprocessing, segmentation, feature extraction, and classification steps.

Outcomes:

On successful completion of this course, students will be able to

- Implement digital image processing fundamentals.
- Apply image enhancement and restoration techniques.

Use image compression, segmentation and features extraction Techniques

CS 19.518 Advanced Database Management Systems Laboratory

Total Hours:26

Credit: 0-0-2-1

Prerequisite: DBMS laboratory

Objectives:

The objectives of this course are:

- To learn object relational databases
- To develop skill to design and analyze cost functions and query optimization
- To implement horizontal and vertical fragmentation

Contents:

- Lab Exercises based on Developing objects, member functions and implementing Object oriented features in SQL
- Lab Exercises based on Developing cost functions and optimizing SQL Queries
- Lab Exercises based on developing strategies for physical database design and tuning in relational systems
- Lab exercises based implementing horizontal fragmentation, vertical fragmentation, total replication and partial replication in distributed database systems.

Outcomes:

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

- Understand total replication and partial replication in distributed database
- Understand Structured Query Language (SQL) with query optimization.

CS 19.563 **Advances in Distributed Computing** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Computer network, Computer Architecture, Distributed System

Objective(s): The objectives of this Course are :

- To learn distributed systems and distributed computing
- To learn synchronous and asynchronous system
- To learn fault tolerant and non-fault tolerant systems.
- To learn synchronous shared memory systems.

1. **Introduction:** Distributed Systems, Cluster Computing, Supercomputing, Cloud Computing. [6]
2. **Synchronous networks and Fault-tolerant Systems:** Leader election in synchronous ring networks. Leader election in rings. Basic computational tasks in general synchronous networks: leader election. Fault-tolerant Systems: Consensus, Link failures: the two generals problem. Process failures (stopping, Byzantine). Algorithms for agreement with stopping and Byzantine failures. Exponential information gathering. [12]
3. **Asynchronous distributed computing and Non-fault-tolerant Systems:** Formal modeling of asynchronous systems using interacting state machines (I/O automata). Proving correctness of distributed algorithms. Non-fault-tolerant Systems: Algorithms for asynchronous networks. Leader election. [10]
4. **Time, clocks, and the ordering of events:** State-machine simulation. Vector timestamps. Stable property detection. Distributed termination. Global snapshots. Deadlock detection, Synchronizers: Synchronizer applications. Synchronous vs. Asynchronous distributed systems. [12]
5. **Asynchronous shared-memory systems and Locking:** The mutual exclusion problem. Mutual exclusion algorithms. Bounds on shared memory for mutual exclusion. Resource allocation. The Dining Philosophers problem. Locking: Locking algorithms, optimistic algorithms, lock-free algorithms, lazy algorithms, Load scheduling and balancing techniques. [12]

Outcome(s): On successful completion the students will be able to:

- Understand distributed algorithms for synchronous and asynchronous distributed computing
- Understand various synchronizer applications with synchronous and asynchronous systems..

Text Books:

[1] Lynch, Nancy, “Distributed Algorithms”, Burlington, MA: Morgan Kaufmann, 1996. ISBN: 9781558603486.

[2] George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems, Concept and Design”, 3rd Edition, Pearson Education, 2005.

Reference Books:

[1] Pradeep K. Sinha, “Distributed Operating Systems: Concepts and Design”, PHI, 2007.

[2] Randy Chow, Theodore Johnson, “Distributed Operating Systems and Algorithm Analysis”, Pearson, 2009.

[3] Mukesh Singhal, Niranjan G. Shivaratri, Niranjan Shivaratri, “Advanced Concepts in Operating Systems”, TMH, 1994.

[4] Sunita Mahajan, Seema Shah, “Distributed Computing”, Oxford University Press,2010

CS 19.565 **Advanced Compiler Construction** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Compiler design, Theory of automata

Objective(s):

- To learn the major conceptual areas of language translation and compiler design.
- To learn the knowledge in code optimization and register allocation techniques
- To extend the knowledge of different parsers.
- To provide practical programming skills necessary for construction of a compiler.

1. **Advanced Issues in Elementary topics:** Compiler, phases and passes, finite state machines, regular expressions and their applications to lexical analysis, implementation of lexical analyzer using FLEX, CFG, bottom-up and top-down parsers, implementation of parsers using YACC. [10]

2. **Symbol Table Structure and Intermediate Representations:** Storage classes, visibility and lifetime. Symbol attributes and symbol table entries, local symbol table management, and global symbol table structure, storage binding and symbolic registers. Intermediate Representations: Issues in designing intermediate languages, High level, medium level, low level and multi-level intermediate languages. Representing all intermediate languages in informal compiler algorithm notation (ICAN). [14]

3. **Code Optimization:** Introduction, early optimization, redundancy elimination. Loop optimization: Induction variable optimizations, bounds checking elimination. Procedure optimization: Tail-call optimization and Tail-recursion optimization, procedure integration, In-line expansion, leaf routine optimization and shrink wrapping. [8]

4. **Register Allocation**: Register allocation and assignment, local methods, graph coloring, priority based graph coloring. Code Scheduling: Instruction Scheduling, Speculative loads and boosting, speculative scheduling, trace scheduling, percolation scheduling. [12]

5. **Inter procedural Analysis and Optimization** Call graph, inter procedural data flow analysis, constant propagation, alias analysis, optimizations and register allocation. [8]

Outcome(s): On successful completion of this course, the students will be able to

- Use lexical analyzer and parser generator tools.
- Generate intermediate code.
- Be familiar with register allocation. and compiler code optimization

Text Books:

[1] Steven S. Muchnik, "Advanced Compiler Design Implementation", Morgan Kauffman publishers, 2005.

[2] Alfred Aho, Monica Lam, Ravi Sethi, Jeffrey Ullman, "Compilers: Principles, Techniques, and Tools", Second edition, Addison-Wesley, 2006.

Reference Books:

[1] Louden, Compiler Construction: Principles and Practice, Cengage Learning, 1st edition, 1997

[2] Fisher, Faraboschi, Young, "Embedded Computing - A VLIW Approach to Architecture, Compilers, and Tools", Morgan Kaufmann, 2005.

CS 19.567 **Bioinformatics** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Computer Programming, DBMS

Objective(s):

- To learn the basics of bioinformatics
- To learn the bioinformatics databases and web techniques
- To extend the knowledge bioinformatics and distributed computing

1. **Introduction to Bioinformatics:** Definition and History of Bioinformatics, Internet and Bioinformatics, Introduction to Data Mining, Applications of Data Mining to Bioinformatics Problems and Applications of Bioinformatics. [6]

2. **Bioinformatics Databases:** Introduction, Nucleotide sequence database - Protein sequence databases, Sequence motif databases, Protein structure databases, other relevant databases. Data storage and retrieval and Interoperability: Flat files, relational, object oriented databases and controlled vocabularies. File Format (Genbank, DDBJ, FASTA, PDB, SwissProt).Introduction to Metadata and search; Indices, Boolean, Fuzzy, Neighboring search. The challenges of data exchange and integration. Ontologies, interchange languages and standardization efforts. General Introduction to XML, UMLS, CORBA, PYTHON. [19]

3. **Sequence Alignments and Visualization:** Introduction to Sequences, alignments and Dynamic Programming; Local alignment and Global alignment (algorithm and example),Pairwise alignment (BLAST and FASTA Algorithm) and multiple sequence alignment (Clustal W algorithm).Methods for presenting large quantities of biological data: sequence viewers (Artemis, SeqVISTA), 3D structure viewers (Rasmol, SPDBv, Chime, Cn3D, PyMol), Anatomical visualization. [16]

4. **Bioinformatics and Web Techniques:** Clustal V, Clustal W 1.7, RasMol, Oligo, Molscript, Tree view, Alscript, Genetic Analysis Software, Phylip. Semantic Web Techniques: Bridging databases, browsing space of information, life sciences identifiers, RDF data representation. [10]

5. **Bioinformatics and Distributed Computing:** Java Distributed computing platform, DPRml, DSEARCH. [5]

Outcome(s): On successful completion of the course, the students will be able to

- Use different bioinformatics databases and web techniques
- Understand sequence alignment algorithms

Text Books:

[1] Dan E. Krane ,"Fundamental Concepts of Bioinformatics", third edition, Pearson Education, 2009.

[2] Claverie, J.M. and Notredame C. ," Bioinformatics for Dummies", Wiley Editor , January 2003.

Reference Books:

[1] Shalini Suri, "Bioinformatics", APH Publishing Corporation, 2006.

CS 19.569 **Advanced Computer Graphics** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Computer Graphics

Objective(s):

- To introduce the students with the principles of animation
- The objective of the course is to equip students with fundamental knowledge and technical competence in the field of computer graphics.
- Provide an understanding of rendering and 3D modeling.
- To learn three dimensional object representation.

1. **3D Transformation and Viewing**: 3D transformation, composite transformation, viewing pipeline, parallel and perspective projection, visible surface detection: classification of visible surface detection algorithms, back face detection, depth buffer method, A-buffer method, scan-line method, depth sorting method. [7]

2. **Animation:** Introduction of computer Animation, Principles of Animation, Keyframing, Kino-dynamics Planning- motion blending, skinning-skeleton animation, rigid skinning ,linear skinning, dual skinning, physics modeling-Physics-based animation, rigid body systems, articulated rigid bodies, forward and inverse dynamics. [7]

3. **Rendering:** Basic concepts of graphics models-illumination& reflection models Shading-Scanning polygons, Gourad and Phong shading , texture mapping- bump & environmental maps , mipmaps , shading languages-vertex shedders and fragment shedders. [12]

4. **Modeling:** Representation, Using homogeneous Coordinate systems, Modeling process, terrain-representation, methods , fault formation & weather, plants modeling , animals modeling. [12]

5. **Surface Representation:** Three dimensional object representation –polygon representations, implicit representation , Beziers ,B-splines-basis function, knot insertion, conversion , rational B-splines (NURBS)-properties and examples , evaluators, subdivision surfaces-coarse mesh and subdivision rule, types of subdivisions , Voxels . [14]

Outcome(s): On successful completion of this course, students will be able to.

- Perform 3D transformation and viewing techniques.
- Understand various styles of animation.
- Acquire knowledge about illumination models, color models and surface rendering.

Text Books:

[1] Peter Shirley, Stephen Robert Marschner, “Fundamentals of Computer Graphics” , A K Peters Limited, 2009.

[2] David F. Rogers, Procedural Elements for Computer Graphics, Tata McGraw Hill Publications, 2nd edition, 2001.

Reference Books:

[1] Shreiner Dave, “Opengl Programming Guide: The Official Guide To Learning Opengl”, Version 2.1, 6/E , Pearson Education India, 2008.

[2] Francis S. Hill, Jr., Stephen M. Kelley , “Computer Graphics: Using OpenGL”, Pearson Prentice Hall, 2008

[3] James D.Foley, Andries van Dam, Steven K. Feiner, John F. Hughes , “Computer Graphics , Principles & Practice”, 3rd Edition, Pearson Prentice Hall, 2013

CS 19.571 **Advances in Algorithm Design** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Data Structures, Design and Analysis of Algorithm

Objective(s):

- To learn the algorithm analysis techniques.
- To become familiar with the different algorithm design techniques.
- To understand the limitations of the algorithms complexity

1. **Pattern matching algorithms:** Finite automata and regular expression, recognition of regular expression patterns, recognition of substrings, Two – way deterministic pushdown automata, position trees and substring identifiers. [11]

2. **Computability theory:** Turing machines, variants of Turing machines, Hilbert‟s problem, decidable languages, the halting problem, Un-decidable problems from language theory, mapping reducibility, the recursion theorem, decidability of logical theories, Turing reducibility, definition of information. [11]

3. **Fast Fourier Transform and Polynomials:** Representation of polynomials, the DFT and FFT algorithms, efficient FFT implementation, the FFT using bit operations, products of polynomials, the Schonhage – Strassen integer multiplication algorithm. [11]

4. **Randomized Algorithm and Parallel Algorithm:** Monte-Carlo algorithm, Las-Vegas algorithm, Game tree evaluation, De-randomization. Models for parallel computation basic techniques; Parallel matrix algorithms: matrix-vector multiplication, matrix multiplication, triangular system solution, sorting networks-parallel merging networks, zero –one principle. [8]

5. **Complexity theory:** Time complexity, the classes P,NP,NP complete and NP hard, Hierarchy theorems, Relativization, Circuit Complexity, approximation algorithms, probabilistic algorithms, circuit complexity, alteration, interactive proof systems, parallel computation, cryptography. [11]

Outcome(s): On the successful completion of the course, the student will be able to:

- Design algorithms for various computing problems.
- Analyze the time and space complexity of algorithms.
- Critically analyze the different algorithm design techniques for a given problem.

Text Books:

[1] Aho Alfred V., Hopcraft John E., Ullman Jefferey D. “The design and analysis of computer algorithms” Pearson education, tenth impression 2012.

[2] Cormen Thomas H. “Introduction to algorithms” PHI, Third edition – 2012.

Reference Books:

[1] MICHAEL SIPSER Sipser Michael, “Introduction to the theory of computation” Thomson press, Third edition 2012.

[2] Brassard G.,Fundamental of Algorithmcs, Prentice-Hall of India, 2003. 4.Horowitz E.,Computer Algorithms, Galgotia Publications, 2002.

CS 19.562 **Mobile Cloud Computing** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Computer Networks, Cloud Computing

Objective(s):

- To learn the cloud services and requirements of cloud
- To become familiar with different cloud models and solutions
- To learn mobile computing architectures

1. **Introduction:** Cloud, Business and IT Perspective, Virtualization, Cloud Services Requirements, Cloud And Dynamic Infrastructure, Cloud Computing Characteristics, Cloud Adoption. [8]
2. **Cloud at a Service:** Internet Software Evolution, Web Services Deliver from the Cloud, Communication-as-a-Service, Infrastructure-as-a-Service, Monitoring-as-a-Service, Platform-as-a-Service, Software-as-a-Service, Building Cloud Network. [8]
3. **Cloud Models and Solutions:** Cloud Models, Cloud Characteristics, Measured Service, Security in a Public Cloud, Public Verses Private Clouds, Cloud Infrastructure Self Service. Cloud Solutions: Cloud Ecosystem, Cloud Business Process Management, Cloud Service Management, Cloud Stack, Computing On Demand, Cloud Sourcing. [16]
4. **Cloud Offerings and Virtualization technology:** Virtual Desktop Infrastructure, Storage Cloud. Cloud Virtualization technology: Virtualization, Virtualization Benefits, Server Virtualization, Hypervisor Management Software, Virtualized Data Center. [12]
5. **Mobile Computing:** Motivation, Applications, Architectures, Platforms For Mobile Cloud Computing, Windows Phone, Android, IPhone, Popular Mobile Apps, 3G, 4G, Bluetooth, Wi-Fi. [8]

Outcomes(s): On the successful completion of the course, the student will be able to:

- To understand the cloud infrastructure
- To understand the different services provided by cloud.
- To understand mobile computing

Text Books:

[1] Kumar Saurabh, “Cloud Computing”, Wiley India, 2011.
[2] Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley: February 2011.

Reference Books:

[1] Haley Beard, “Cloud Computing Best Practices for Managing and Measuring Processes for On Demand computing applications and data Centers in the Cloud with SLAs” , Emereo Pty Limited, July 2008.
[2] Asoke Talkukder, Roopa R Yavagal, “Mobile Computing – Technology, Applications and Service Creation”, Tata McGraw Hill, 2007
[3 Raj Kamal, “Mobile Computing”, Oxford University Press, 2007.
[4] Michael Miller, “Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online”, 1st Edition, Publisher: Pearson, 2008.

CS 19.564 **Machine Intelligence and Robotics** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Artificial Intelligence

Objective(s):

- To learn artificial intelligence and knowledge representation
- To become familiar with knowledge organization and control strategies
- To learn robotics and robot programming

1. **Introduction:** Importance of AI, Knowledge Based System, Knowledge organization & manipulation, Conceptual Introduction to LISP and other AI programming Languages. [10]

2. **Knowledge Representation:** Syntax, Semantics, Inference Rules, Non-deductive Inference methods, and representations using rules. Forward chaining and backward chaining. Fuzzy Logic & Natural languages computations. Probabilistic Reasoning. Object Oriented Representation. [10]

3. **Knowledge Organization & Manipulation:** Search & control strategies, matching techniques, knowledge organization & management, Genetic Algorithms based search techniques. [10]

4. **Robotics:** Types of Robots, spatial transformation and kinematics of open chain linkages. Mobile robots, Actuators, sensors, programming and control. Applications - motion planning, grasping and industrial automation. [12]

5. **Robot Programming:** Languages, characteristics of Languages, Position Specification, Motion Specification. Robot Program synthesis, Programming solution using VAL Robot programming Language, Artificial Intelligence, search strategies, Heuristic search, Rule based problem solving, Knowledge Representation, Robot programming using LEGO robots. [10]

Outcomes(s): On the successful completion of the course, the student will be able to:

- To understand robot programming using LEGO NXT EV3
- To understand VAL robot programming language
- • To understand knowledge representation and AI Programming Languages.

Text Books:

[1] R.D. Klafter, Chemieleskio, T.A. and Negin M., "Robotics Engineering an Integrated approach", Prentice Hall , 1989.
[2] M. Nagenevtsky, "Artificial Intelligence –A guide to intelligent systems", Addison – Wesley , 2005.

Reference Books:

[1] K.S.FU, R.C. Gonzalez, and C.S.G.Lee., "Robotics control , Sensing, Vision ,and Intelligence" , Prentice Hall , 1987.
[2] J.J. Craig, "Introduction to Robotics", Addison Wesley Publishers, 2005.
[3] G. Bekey, "Autonomous Robots", MIT Press, 2005.

CS 19.566 **Natural Language Processing** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Artificial Intelligence

Objective(s):

- To learn language modeling, syntactic analysis and parsing
- To learn semantic analysis and parsing
- To learn robotics and robot programming

1. **Language Modelling and Syntactic Analysis :** Definition, Origin of NLP, Language and Knowledge, The challenges of NLP, Language and Grammar, NLP application, Successful early NLP systems, Information retrieval. Various grammar-based language models, statistical language models. [16]

   Word Level Analysis and Syntactic Analysis: Introduction, Regular Expressions, Finite State Automata, Morphological Parsing, Spelling Error Detection and correction, words and words classes, Part-of-speech tagging . Context Free grammar, Constituency, Parsing, Probability Parsing.

2. **Semantic Analysis and Discourse Processing:** Introduction, Meaning Representation, Lexical Semantics, Ambiguity, word sense Disambiguation. Cohesion, Reference Resolution, Discourse Coherence and Structure. [8]

3. **Natural Language generation and Machine Translation:** Introduction, Architecture of NLG systems, Generation task and representation, Problems in Machine Translations, Machine Translation approaches, Direct Machine Translation, Rule based machine translation, corpus-based MT, Semantic or knowledge based MT systems. [10]

4. **Application and Lexical Resources:** Speech Recognition and Synthesis, Information Retrieval Systems, Information Extraction, Automatic Text Summarization, Question-Answering; Resources**:** WordNet, FrameNet, Stemmers, Part-of Speech Tagger. [10]

5. **NLP with Python:** Introduction, Processing Raw Text, Writing Structured Program, Categorizing and Tagging words, learning to Classify Text, Analyzing Sentence Structure. [8]

Outcome(s): On the successful completion of the course, the student will be able to:

- To understand Implementation of Natural language processing using Python
- To understand VAL robot programming language
- To understand knowledge representation and AI Programming Languages.

Text Books:

[1] Steven Bird, Ewan Klein and Edward Loper, "Natural Language Processing with Python", O" Reilly Media , 2009.

Reference Books:

[1] U. S. Tiwary and Tanveer Siddiqui, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.
[2] Christopher D. Manning and Hinrich Schutze," Foundation of Statistical Natural Language Processing", ISBN 026213360

CS 19.568

Advanced Software Engineering

Total Lectures: 52

Credit: 4-0-0-4

Prerequisite: Software Engineering

Objective(s):

- To understand the software engineering approaches, software development process models, project planning, software size estimation, software effort and cost estimation and risk management.
- To acquire knowledge on implementation issues in software design..
- To learn coding style guidelines and standard and software testing, different type of testing strategies and their application in real time system.
- To study object oriented system analysis, design and testing.

1. **Introduction:** Software life cycle models, Waterfall, Prototype, Evolutionary and Spiral Models, Overview of Quality Standards like ISO 9001, SEI – CMM, Software Requirements analysis & specifications, Software Project Management. [10]
2. **Fundamental issues in software design:** Goodness of design, cohesions, coupling, Classification of Cohesiveness & Coupling, Function-oriented design: structured analysis and design. [8]
3. **Software Testing:** Testing process, Design of test cases, functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path Testing, Data flow and mutation testing, Unit Testing, Integration and System Testing, Debugging, Alpha & Beta Testing, Regression Testing, Testing Tools & Standards. [8]
4. **Object Oriented System Analysis. Design and Testing:** Object oriented and object basics, object oriented system development cycle, process framework of OOSAD, unified approach UML, UML Diagrams: class diagram, use-case diagram, behavior diagram, OOAP, OOD and OODBMS.
   Object-Oriented Testing: Testing Object Oriented Systems overview. Broadening the View of Testing, Testing OOA and OOD Models, Object-Oriented Testing Strategies, Test Case Design for OO Software , Testing Methods Applicable at the Class Level. [16]
5. **Advanced Topics in Software Engineering:** Overview of Formal Methods, Cleanroom Software Engineering, Component-Based Software Engineering, Client-Server Software Engineering, Web Engineering, Reengineering, Computer-Aided Software Engineering, and the Road Ahead. [10]

Outcome(s): On successful completion of this course, students will be able to

- Demonstrate knowledge of the distinction between critical and non-critical systems and ability to manage a project including planning, scheduling and risk assessment/management.
- Demonstrate proficiency of different life cycle models.
- Identify specific components of a software design that can be targeted for reuse.
- Demonstrate proficiency in software development cost estimation, software testing plan and advanced topics of software engineering.

Text Books:

[1] R. S. Pressman, “Software Engineering – A practitioner‟s approach”, 5th Ed., McGraw Hill Int. Ed., 2001.
[2] Ian Sommerville, “Software Engineering”, Addison. Wesley, 1999.

Reference Books:

[1] K. K. Aggarwal & Yogesh Singh, “Software Engineering”, New Age International, 2001.
[2] Stephen R. Schach, “Classical & Object Oriented Software Engineering”, IRWIN, 1996.
[3] P. Jalote, “An Integrated approach to Software Engineering”, Narosa, 1991.
[4] Booch, Rumbaugh and Jacobson “The Unified Modeling Language User Guide”, Addison-Wesley 2005.

CS 19.570 **Mathematical Modeling** **Total Lectures: 52**

Credit: 4-0-0-4

Prerequisite: Engineering Mathematics

Objective(s):

- To acquire knowledge about basics of various mathematical models
- To learn methods of model fitting
- To study modeling with differential equations

1. **Introduction :** Basics of modeling, mathematical models, Modeling change with difference equations, Approximate change with difference equations, solutions to dynamical systems, systems of difference equations. Modeling using proportionality, modeling using geometric similarity. [12]

2. **Model fitting and Experimental modeling** : Fitting models to data graphically, analytic method of model fitting, applying the least square criterion, choosing a best model, one term models , high order polynomial models, smoothing of low order polynomial models, cubic spline models. [10]

3. **Simulation modeling:** Introduction, simulating deterministic behavior: area under a curve, generating random numbers, simulates probabilistic behavior, queuing models. Introduction to some simulators. [10]

4. **Discrete modeling:** Probability modeling with discrete systems, linear regression, an overview of discrete optimization modeling, linear programming: geometric and algebraic solutions, simplex method. [10]

5. **Modeling with differential equations:** Introduction, population growth, graphical solutions of autonomous differential equations, numerical approximation methods, graphical solutions of autonomous systems of First order differential equations. [10]

Outcome(s): On successful completion of this course, students will be able to

- Understand simulation modeling and discrete optimization modeling
- Understand queueing models and low order polynomial models

Text Books:

[1] Giordano, Weir, and Fox, “A First Course in Mathematical Modeling”, 4th edition, Brooks/Cole Thomson Learning, 2009.
[2] Randall James Swift, “A Course in Mathematical Modeling”, Cambridge University Press, 1999.

Reference Books:

[1] Edward A. Bender, “An Introduction to Mathematical Modeling”, Courier Dover Publications, 2000.
[2] Neville D. Fowkes, John J. Mahony, “An introduction to mathematical modeling”, Wiley, 1994.
[3] Jerry Banks, John S. Carson II Barry L. Nelson David M. Nicol,” discrete event system simulation ”, Prentice Hall, fourth edition , 2004