AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M. Tech - Computer Science and Engineering

Program Outcomes

PO1: Independently carry out research/investigation and development work to solve practical problems.

PO2: Write and present a substantial technical report/document.

PO3: Demonstrate a degree of mastery in the area of Computer Science and Engineering.

Program Specific Outcomes:

PSO1: Demonstrate expertise in specialized domains such as Artificial Intelligence, Data Science, Cyber security, and Cloud Computing, contributing to cutting-edge Technological innovations.

PSO2: Formulate research problems, design experiments, use modern tools, publish and protect outcomes, and translate results into deployable prototypes or technology transfers.

AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Structure

M.Tech (CSE) I–Semester

S. No	Course Title	L	T	P	C
1	**Program Core–1** Data Structures and Algorithm Analysis	3	1	0	4
2	**Program Core–2** Advanced Data Warehousing and Data Mining	3	1	0	4
3	**Program Core–3** Mathematical Foundations of Computer Science	3	1	0	4
4	**Program Elective–I** 1. Image Processing 2. Soft computing 3. Advanced Computer Networks 4. Advanced Software Engineering	3	0	0	3
5	**Program Elective–II** 1. Advanced Compiler Design 2. Time Series Analysis 3. High Performance Computing 4. Agile Methodologies	3	0	0	3
6	**Laboratory–1** Data Structures and Algorithm Analysis Lab	0	1	2	2
7	**Laboratory–2** Advanced Data warehousing and Data Mining Lab	0	1	2	2
8	Seminar-I	0	0	2	1
	TOTAL				**23**

AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

List of Professional Elective Courses in I Semester (Electives– I & II)

S. No.	Course Title
1.	1. Image Processing 2. Soft computing 3. Advanced Computer Networks 4. Advanced Software Engineering 5. Time Series Analysis 6. High Performance Computing 7. Agile Methodologies 8. Advanced Compiler Design 9. Any minimum 12 weeks MOOCS/NPTEL courses suggested by BOS

AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

M.Tech (CSE) II– Semester

Sl. No.	Course Title	L	T	P	C
1	**Program Core–4** Machine Learning	3	1	0	4
2	**Program Core–5** Cryptography and Network Security	3	1	0	4
3	**Program Core–6** Introduction to Quantum Computing	3	1	0	4
4	**Program Elective – III** 1. Feature Engineering 2. Generative AI 3. Design Patterns 4. Natural Language Processing	3	0	0	3
5	**Program Elective – IV** 1. Block Chain Technologies 2. Adhoc Sensor Networks 3. DevOps 4. Secure Coding	3	0	0	3
6	**Laboratory– 3** Machine Learning Lab	0	1	2	2
7	**Laboratory– 4** Natural Language Processing Lab	0	1	2	2
8	Seminar–II	0	0	2	1
	TOTAL				**23**

AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

List of Professional Elective Courses in II Semester (Electives III & IV)

S. No.	Course Title
1	1. Feature Engineering 2. Generative AI 3. Adhoc Sensor Networks 4. Principles of Network Security 5. Block Chain Technologies 6. DevOps 7. Secure Coding 8. Design Patterns 9. Any minimum 12 weeks MOOCS/NPTEL courses suggested by BOS

AMRITASAIINSTITUTEOFSCIENCEANDTECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada,
Recognized by UGC under 2(f) & 12(B) of 1956 Act.
AnISO9001:2015Certifiedinstitution, Accredited by NAAC, Paritala,
Kanchikacherla, Krishna Dist., Andhra Pradesh521180.
www.amritasai.org.in,Phone:08678-2477777

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGs

M.Tech (CSE) –III Semester

S. No	Course Title	L	T	P	C
1	Research Methodology and IPR / *Swayam12 week MOOC course – RM&IPR*	3	0	0	3
2	Summer Internship / Industrial Training (8-10 weeks)*	-	-	-	3
3	Comprehensive Viva*	-	-	-	2
4	Dissertation Part–A	-	-	20	10
	TOTAL		**-**		**18**

*Student shall take up the internship during summer/year break and assessment will be done in 3rdSem- Internal Assessment

*Comprehensive Viva shall be conducted for courses completed till second semester.- Internal Assessment

*Dissertation Part-A- Internal Assessment

M. Tech (CSE) – IV Semester

S. No.	Course Title	L	T	P	C
1	Dissertation Part – B*	-	-	32	16
	TOTAL	**-**	**-**	**32**	**16**

*Dissertation Part-B- External Assessment

M Tech CSE- I Semester		PC	L	T	P	C
			3	1	0	4
	Data Structures and Algorithm Analysis					

Course Objectives: The main objectives of introducing this course are

1. Introduce basic concepts and techniques of data warehousing and data mining.
2. Examine the types of the data to be mined and apply pre-processing methods on raw data.
3. Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.

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

CO1	Analyze algorithms for algorithm correctness and efficiency
CO2	Develop and implement advanced abstract data types (ADTs) and data structures to solve complex computational problems
CO3	Demonstrate various searching, sorting and hash techniques and be able to apply and solve problems of real life
CO4	Apply data structures including linked lists, trees, heaps, and graphs to implement computational solutions
CO5	Compare various search trees and find solutions for IT related problems

SYLLABUS	
UNIT-I:	**Introduction to Data Structures**
Singly Linked Lists, Doubly Linked Lists, Circular Lists-Algorithms, Stacks and Queues- Algorithm Implementation using Linked Lists.	
UNIT-II:	**Searching**
Linear and Binary, Search Methods, Sorting- Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Trees- Binary trees, Operations-Insertion, Deletion, Properties, Representation and Traversals (DFT, BFT), expression Trees (Infix, prefix, postfix), Graphs- Basic Concepts,	

Storage structures and Traversals	
UNIT-III:	**Dictionaries**
Dictionaries, ADT, The List ADT, Stack ADT, Queue ADT, Hash Table Representation, Hash Functions, Collision Resolution-Separate Chaining, Open Addressing- Linear Probing, Double Hashing.	
UNIT-IV:	**Priority queues**
Definition, ADT, Realising a Priority Queue Using Heaps, Definition, Insertion, Deletion, Search Trees- Binary Search Trees, Definition, ADT, Implementation, Operations- Searching, Insertion, Deletion.	
UNIT-V:	**Search Trees**
AVL Trees, Definition, Height of AVL Tree, Operations- Insertion, Deletion and Searching. Introduction to Red-Black and Splay Trees, B-Trees, Height of B-Tree, Insertion, Deletion and Searching, Comparison of Search Trees.	
Text Books 1. Data Structures: A Pseudocode Approach with C, 2 nd Edition, Richard F.Gilberg, Behrouz A. Forouzon, Cengage Learning, 2004 2.Data Structures, Algorithms and Applications in java, 2 nd Edition, Sartaj Sahni, University Press/Orient BlackSwan, 2005	
Reference Books 1. Data Structures And Algorithm Analysis, 2nd Edition, Mark Allen Weiss, Pearson, 2002 2. Data Structures And Algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning, 2005 3. C and Data Structures: A Snap Shot Oriented Treatise Using Live Engineering Examples, 1 st Edition, N.B.Venkateswarulu, E.V. Prasad, S Chand & Co, 2009 4. Classic Data Structures, 2nd Edition, Debasis Samantha, PHI Learning, 2009	

M.Tech I Semester		PC	L	T	P	C
			3	1	0	4
	ADVANCED DATA WAREHOUSING AND DATA MINING					

Course Objectives: The main objectives of introducing this course are

1. Understand Data Warehousing Concepts
2. Apply Data Mining Methodologies
3. Perform Data Preprocessing and Exploration
4. Develop and Evaluate Classification Models
5. Apply Predictive Analytics Concepts

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

CO1	Demonstrate the architecture, modeling techniques, and implementation strategies of data warehouses and OLAP systems, including modern cloud-based approaches
CO2	Apply statistical and visualization techniques to describe datasets and perform data preprocessing tasks such as cleaning, integration, reduction, and transformation.
CO3	Develop and evaluate classification models using decision trees, Bayesian classifiers, and rule-based methods for solving predictive analytics problems.
CO4	Apply algorithms such as Apriori, FP-Growth, and sequential pattern mining to discover meaningful associations and patterns in data.
CO5	Solve data clustering problems by implementing techniques such as K-means, hierarchical clustering, and DBSCAN for text, spatial, and graph data

SYLLABUS	
UNIT-I:	**Data Warehousing and Online Analytical Processing**
Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data	
UNIT-II:	**Data Objects and Attributes**
Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. (Text Book- 1)	
UNIT-III:	**Classification**
Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection. (Text Book- 2)	
UNIT-IV:	**Association Analysis**
Problem Definition, Frequent Itemset Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm, Sequential Patterns: Preliminaries, Sequential Pattern Discovery (Text Book- 2)	
UNIT-V:	**Cluster Analysis**
Clustering techniques, Different Types of Clusters; K- means: The Basic K-means Algorithm, K- means Additional Issues, Bisecting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. Mining rich data types: Mining text data, Spatial-temporal data, Graph and networks. (Text Book- 2)	
Text Books 1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011. 2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.	
Reference Books 1. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher. 2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press,2013	
Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse; Data Mining Methodologies: CRISP-DM and SEMMA, Comparison of Data Mining Methodologies. Statistical Limits on Data Mining, Introduction to Predictive Analytics, Technologies, Applications, Major issues.	

M Tech I Semester		PC	L	T	P	C
			3	1	0	4
	MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE					

Course Objectives: The main objectives of introducing this course are

1. To apply the basic rules and theorems of probability theory such as Baye's Theorem, to determine probabilities that help to solve engineering problems and to determine the expectation and variance of a random variable from its distribution.
2. Able to perform and analyze of sampling, means, proportions, variances and estimates the maximum likelihood based on population parameters.
3. To learn how to formulate and test hypotheses about sample means, variances and proportions and to draw conclusions based on the results of statistical tests.

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

CO1	Explain foundational concepts in probability, statistics, algebraic structures, and graph theory.
CO2	Apply statistical and algebraic techniques to solve real-world problems involving data analysis, number theory, and modular arithmetic.
CO3	Analyze different types of distributions, sampling methods, and graph structures to determine relationships and draw inferences.
CO4	Evaluate statistical hypotheses, estimator properties, and algorithmic outcomes using appropriate significance tests and decision criteria.
CO5	Construct and formulate mathematical models using probability distributions, algebraic systems, and graph-theoretical approaches for applications in data science and computing.

SYLLABUS	
UNIT-I:	**Basic Probability and Random Variables:**
Random Experiments, Sample Spaces Events, the Concept of Probability the Axioms of Probability, Some Important Theorems on Probability Assignment of Probabilities, Conditional Probability Theorems on Conditional Probability, Independent Events, Bayes Theorem or Rule. Random Variables, Discrete Probability Distributions, Distribution Functions for Random Variables, Distribution Functions for Discrete Random Variables, Continuous Random Variables	

UNIT-II:	Sampling and Estimation Theory

Population and Sample, Statistical Inference Sampling With and Without Replacement Random Samples, Random Numbers Population Parameters Sample Statistics Sampling Distributions, Frequency Distributions, Relative Frequency Distributions, Computation of Mean, Variance, and Moments for Grouped Data. Unbiased Estimates and Efficient Estimates Point Estimates and Interval Estimates. Reliability Confidence Interval Estimates of Population Parameters, Maximum Likelihood Estimates.

UNIT-III:	Tests of Hypothesis and Significance

Statistical Decisions Statistical Hypotheses. Null Hypotheses Tests of Hypotheses and Significance Type I and Type II Errors Level of Significance Tests Involving the Normal Distribution One-Tailed and Two- Tailed Tests P Value Special Tests of Significance for Large Samples Special Tests of Significance for Small Samples Relationship between Estimation Theory and Hypothesis Testing, Operating Characteristic Curves. Power of a Test Quality Control Charts Fitting Theoretical Distributions to Sample Frequency Distributions, The Chi- Square Test for Goodness of Fit Contingency Tables Yates' Correction for Continuity Coefficient
of Contingency.

UNIT-IV:	Algebraic Structures and Number Theory

Algebraic Systems, Examples, General Properties, Semi Groups and Monoids, Homomorphism of Semi Groups and Monoids, Group, Subgroup, Abelian Group, Homomorphism, Isomorphism. Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem)

UNIT-V:	Graph Theory

Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Planar Graphs, Euler's Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems without Proofs).

Text Books

1. Foundation Mathematics for Computer Science,1st edition, John Vince, Springer,2015
2. Probability & Statistics, 3rd Edition, Murray R. Spiegel, John J. Schiller and R. Alu Srinivasan, Schaum's Outline Series, Tata McGraw-Hill Publishers, 2018
3. Probability and Statistics with Reliability,2nd edition, K. Trivedi, Wiley, 2011

4.Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, H. Rosen, Tata McGraw Hill, 2003

Reference Books

1. Probability and Computing: Randomized Algorithms and Probabilistic Analysis,1 st edition, M. Mitzenmacher and E. Upfal,2005
2. Applied Combinatorics,6th edition, Alan Tucker, Wiley,2012

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	IMAGE PROCESSING (PROGRAM ELECTIVE-I)					

Course Objectives: The main objectives of introducing this course are

1. Understand the foundational concepts of digital image processing, including image representation, acquisition, sampling, and quantization.
2. Interpret and explain common image processing techniques such as filtering, enhancement, and noise reduction in both spatial and frequency domains.
3. Understand and describe the purpose and functioning of advanced image processing tasks such as segmentation, morphological operations, and compression.

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

CO1	Explain basic principles of digital image processing.
CO2	Apply algorithms to perform basic image processing tasks such as noise removal and image enhancement.
CO3	Develop and implement algorithms for advanced image analysis, including image compression and segmentation.
CO4	Apply evaluation methods to assess the performance of image processing algorithms and systems.
CO5	Utilize performance evaluation methods to assess image segmentation, morphological operations, and feature extraction techniques for image analysis tasks

SYLLABUS	
UNIT-I:	**Introduction:**
Fundamental steps in Image Processing System, Components of Image Processing System, Elements of Visual Perception, Image Sensing and acquisition, Image sampling & Quantization, Basic Relationship between pixels. Image Enhancement Techniques: Spatial Domain Methods: Basic grey level transformation, Histogram equalization, Image subtraction, image averaging	
UNIT-II:	**Spatial filtering**
Smoothing, sharpening filters, Laplacian filters, Frequency domain filters, Smoothing and sharpening filters, Homomorphism is filtering. Image Restoration & Reconstruction: Model of Image Degradation/restoration process, Noise models, Spatial filtering, Inverse filtering, Minimum mean square Error filtering, constrained least square filtering, Geometric mean filter, Image reconstruction from projections. Color Fundamentals, Color Models, Color Transformations.	

UNIT-III:	**Image Compression**	
Redundancies- Coding, Interpixel, Psycho visual; Fidelity, Source and Channel Encoding, Elements of Information Theory; Loss Less and Lossy Compression; Run length coding, Differential encoding, DCT, Vector quantization, Entropy coding, LZW coding; Image Compression Standards-JPEG, JPEG 2000, MPEG; Video compression.		
UNIT-IV:	**Wavelet Based Image Compression**	
Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous, Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding		
UNIT-V:	**Image Segmentation:**	
Discontinuities, Edge Linking and boundary detection, Thresholding, Region Based Segmentation, Watersheds; Introduction to morphological operations; binary morphology- erosion, dilation, opening and closing operations, applications; basic gray-scale morphology operations; Feature extraction; Classification; Object recognition. Digital Image Watermarking: Introduction, need of Digital Image Watermarking, applications of watermarking in copyright protection and Image quality analysis		
Text Books 1. Digital Image Processing n2nd ed. Gonzalez, R.C. and Woods, R.E. India: Person Education,2009		
Reference Books 1. Digital Image Processing. John Wiley, Pratt, W. K, Fourth Edition-2001. 2. Digital Image Processing, Jayaraman, S., Veerakumar, T. and Esakkiranjan, S.,Tata McGraw-Hill,Edition-3,2009.		

M. Tech I Semester		PE	L	T	P	C
			3	0	0	3
	SOFT COMPUTING (PROGRAM ELECTIVE-I)					

Course Objectives: The main objectives of introducing this course are

1. To introduce the concepts in Soft Computing such as Artificial Neural Networks, Fuzzy logic-based systems, genetic algorithm-based systems and their hybrids

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

CO1	Understand the principles of soft computing, including their biological and mathematical foundations.
CO2	Apply neural network models to implement pattern recognition and learning tasks..
CO3	Explain fuzzy set theory and fuzzy logic concepts.
CO4	Develop and evaluate fuzzy inference systems and understand the design and implementation of neuro-fuzzy hybrid systems.
CO5	Apply genetic algorithms for optimization problems, including coding, selection, crossover, mutation, and stopping criteria, and explore their integration with neural and fuzzy systems.

SYLLABUS	
UNIT-I:	**Introduction**
Introduction to Soft Computing, Artificial neural networks, biological neurons, Basic models of artificial neural networks, Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network	
UNIT-II:	**Perceptron Networks**
Perceptron networks, Learning rule, Training and testing algorithm, Adaptive Linear Neuron, Back propagation Network, Architecture, Training algorithm.	
UNIT-III:	**Representation of Knowledge**
Fuzzy logic, fuzzy sets, properties, operations on fuzzy sets, fuzzy relations, operations on fuzzy relations, Fuzzy membership functions, fuzzification, Methods of membership, value assignments, intuition, inference, rank ordering, Lambda –Cuts for fuzzy sets , Defuzzification methods.	

UNIT-IV:	Logic Concepts
Truth values and Tables in Fuzzy Logic, Fuzzy propositions, Formation of fuzzy rules, Decomposition of rules, Aggregation of rules, Fuzzy Inference Systems, Mamdani and Sugeno types, Neuro-fuzzy hybrid systems, characteristics, classification	
UNIT-V:	**Expert Systems**
Introduction to genetic algorithm, operators in genetic algorithm, coding, selection, crossover, mutation, stopping condition for genetic algorithm flow, Genetic-neuro hybrid systems, Genetic Fuzzy rule based system	

Text Books

1. S. N. Sivanandam and S. N. Deepa, Principles of soft computing–John Wiley &Sons,2007.
2. Timothy J. Ross, Fuzzy Logic with engineering applications, John Wiley & Sons, 2016.

Reference Books

1. N.K. Sinhaand M.M. Gupta,Soft Computing & Intelligent Systems:Theory &Applications-Academic Press /Elsevier. 2009
2. Simon Haykin, Neural Network-A Comprehensive Foundation-Prentice Hall International, Inc.1998
3. Driankov D., Hellendoorn H.and Reinfrank M.,An Introduction to Fuzzy Control Narosa Pub., 2001.
4. BartKosko, Neural Network and Fuzzy Systems-Prentice Hall,Inc.,Englewood Cliffs, 1992
5. Goldberg D.E, Genetic Algorithms in Search , Optimization , and Machine LearningAddison Wesley, 1989

Online Learning Resources:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	ADVANCED COMPUTER NETWORKS (PROFESSIONAL ELECTIVE-I)					

Course Objectives: The main objectives of introducing this course are

1. To Provide basic understanding of Computer networks starting with OSI Reference Model, Protocols at different layers with special emphasis on IP, TCP & UDP and Routing Algorithms
2. To understand CSMA/CD, TCP/IP implementation, LANs/WANs, internetworking technologies, Routing and Addressing.
3. To introduce assembly language programming concepts.
4. Provide the mathematical background of routing protocols.
5. To develop familiarity with current research problems and research methods in advance computer networks.

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

CO1	Illustrate reference models with layers, protocols, and interfaces.
CO2	Demonstrate routing algorithms, subnetting, and addressing in IPv4 and IPv6.
CO3	Analyze basic network protocols and their use in network design and implementation.
CO4	Explain the concepts related to wireless networks such as WLANs, WiMAX, IEEE 802.11, cellular and satellite systems.
CO5	Solve networking challenges by applying emerging trends such as MANETs and Wireless Sensor Networks

SYLLABUS	
UNIT-I:	**Network layer**
Network Layer design issues: store-and forward packet switching, services provided transport layers, implementation connection less services, implementation connection oriented services, comparison of virtual – circuit and datagram subnets, Routing Algorithms-shortest path routing, flooding, distance vector routing, link state routing, Hierarchical routing, congestion control algorithms : Approaches to congestion control, Traffic aware routing, Admission control, Traffic throttling, choke Packets, Load shedding, Random early detection, Quality of Service, Application requirements, Traffic shaping, Leaky and Token buckets.	

UNIT-II:	**Internetworking and IP protocols**
How networks differ, How networks can be connected, internetworking, tunneling, The network layer in the internet, IPV4 Protocol, IP addresses, Subnets, CIDR, classful and Special addressing, network address translation (NAT),IPV6 Address structure address space, IPV6 Advantages, packet format, extension Headers, Transition from IPV4 to IPV6 , Internet Control Protocols- IMCP, ARP, DHCP.	
UNIT-III:	**Transport Layer Protocols**
Introduction, Services, Port numbers, User Datagram Protocol: User datagram, UDP services, UDP Applications, Transmission control Protocol: TCP services, TCP features, Segment, ATCP connection, State transition diagram, Windows in TCP, Flow control and error control, TCP Congestion control, TCP Timers, SCTP: SCTP services SCTP features, packet format, An SCTP association, flow control, error control.	
UNIT-IV:	**Wireless LANS**
Introduction, Architectural comparison, Access control, The IEEE 802.11 Project: Architecture, MAC sub layer, Addressing Mechanism, Physical Layer, Bluetooth: Architecture, Bluetooth Layers Other Wireless Networks: WIMAX: Services, IEEE project 802.16,Layers in project 802.16, Cellular Telephony: Operations, First Generation (1G), Second Generation (2G), Third Generation (3G), Fourth Generation (4G), Satellite Networks: Operation, GEO Satellites, MEO satellites, LEO satellites	
UNIT-V:	**Emerging trends in Computer Networks**
Mobile Computing: Motivation for mobile computing, Protocol stack issues in mobile computing environment, mobility issues in mobile computing, security issues in mobile networks, MOBILE Ad Hoc Networks: Applications of Ad Hoc Networks, Challenges and Issues in MANETS, MAC Layer Issues Routing Protocols in MANET, Transport Layer Issues, Ad hoc Network Security **Wireless Sensor Networks:** WSN functioning, Operating system support in sensor devices, WSN characteristics, sensor network operation, Sensor Architecture: Cluster management, Wireless Mesh Networks: WMN design, Issues in WMNs, Computational Grids, Grid Features, Issues in Grid construction design, Grid design features,P2P Networks: Characteristics of P2P Networks, Classification of P2P systems, Gnutella, BitTorrent, Session Initiation Protocol(SIP) , Characteristics and addressing, Components of SIP, SIP establishment, SIP security.	
Text Books 1. Data communications and networking 4th edition Behrouz A Fourzan,TMH- 2007 2.Computer networks 4th edition Andrew S Tanenbaum, Pearson,2012 3.Computer networks, Mayank Dave,CENGAGE, First edition.2012	

Reference Books

1. Computer networks, A system Approach, 5thed, Larry L Peterson and Bruce S Davie, Elsevier-2012.

M Tech I Semester		PE-1	L	T	P	C
			3	0	0	3
	ADVANCED SOFTWARE ENGINEERING (PROGRAM ELECTIVE-I)					

Course Objectives: The main objectives of introducing this course are

1. Understand the phases of the software development lifecycle and analyze common process models such as Waterfall, the Unified Process, and Agile methodologies.
2. Apply software engineering practices including requirements analysis, specification, code analysis, debugging, and testing to develop reliable software systems.
3. Design effective software solutions using appropriate software design techniques and best practices in software engineering.

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

CO1	Demonstrate software process, various models and Agile methodologies
CO2	Analyze and Specify software requirements through a SRS documents
CO3	Design and Plan software solutions to problems
CO4	Analyze the importance of Quality assurance and design, implement, and execute test cases at the Unit level.
CO5	Develop and execute integration-level test cases, and analyze the role of various metrics in software testing.

SYLLABUS	
UNIT-I:	**Software and Software Engineering**
Nature of software, Software Process, Software Engineering Practice. Process Models: Generic process model, defining a framework activity, identifying task set, process assessment and improvement, perspective process models Agility and process: Agility, Agile process, Scrum, other Agile frameworks, recommended process model	
UNIT-II:	**Human aspects of Software Engineering**
Characteristics and psychology of Software Engineer, software team, team structure. Principles that guide practice: core principles, principles that guide each framework activity. Understanding Requirements: Requirements engineering, establishing groundwork, requirements gathering, developing use cases, building analysis model, negotiating requirements, requirements monitoring, validating Requirements modeling: requirements analysis, class-based modeling, functional modeling, behavioral modeling.	

UNIT-III:	Design

Design process, design concepts, design model Architectural design: Software architecture, architectural styles, architectural design assessing alternative architectural designs.

User experience design: elements, golden rules, User interface analysis and design, user experience analysis, user interface design, design evaluation, usability and accessibility

Design for mobility: mobile development life cycle, mobile architecture, web design pyramid, mobility and design quality, best practices.

UNIT-IV:	Quality

Software quality, quality dilemma, achieving software quality Reviews: Review metrics, Informal reviews, Formal technical reviews.

Software Quality Assurance: elements, SQA process, Product characteristics, SQA tasks, goals and metrics, statistical software quality assurance, software reliability, ISO 9000 quality standards, SQA plan.

Software testing: strategic approach to software testing, planning and recordkeeping, test case design, white box testing, black box testing, object oriented testing.

UNIT-V:	Software testing

Integration level: Software testing fundamentals, integration testing, regression testing, integration testing in OO context, validation testing.

Software testing- testing for mobility: mobile testing guidelines, testing strategies, User experience testing issues, web application testing, Web testing strategies, security testing, performance testing.

Software metrics and analytics: software measurement, software analytics, product metrics, metrics for testing, metrics for maintenance, process and project metrics, software measurement, metrics for software quality.

Text Books

1. "Software Engineering, A practitioner's Approach", Roger S. Pressman, Bruce R. Maxim, 9th Edition, Tata McGraw-Hill.
2. "Software Engineering", Ian Sommerville, 9th edition, Pearson education

Reference Books

1. Software Engineering: A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	ADVANCED COMPILER DESIGN (Program Elective – II)					

Course Objectives: The main objectives of introducing this course are

1. Understand the fundamental phases of a compiler and the theory behind lexical, syntax, and semantic analysis, as well as the construction of parsing techniques.
2. Apply principles of syntax-directed translation, intermediate code generation, and type checking for various programming constructs.
3. Analyze and optimize intermediate code using control flow and data-flow analysis techniques and generate efficient machine-level code using standard code generation strategies.

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

CO1	Explain the role and structure of each phase of a compiler
CO2	Utilize lexical analyzer tools to design and implement efficient tokenization solutions
CO3	Analyze and compare different parsing techniques and apply appropriate error recovery methods.
CO4	Construct syntax-directed definitions and translation schemes to generate intermediate code.
CO5	Develop and optimize code using data-flow analysis, loop optimization, and register allocation, and implement a simple code generator for a high-level language.

SYLLABUS	
UNIT-I:	**Lexical Analysis**
Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.	
UNIT-II:	**Syntax Analysis**
The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring, Top Down Parsing: Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing	
UNIT-III:	**Bottom Up Parsing**
Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing,	

SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parses, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers

UNIT – IV	**Syntax Directed Translation**

Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax Directed Translation, Syntax- Directed Translation Schemes, Implementing L-Attributed SDD's. Intermediate Code Generation: Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

UNIT-V:	**Run Time Environments**

Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays, Code Optimization: The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization, Code Generation: Issues in the Design of a Code Generator, Object Code Forms, Code Generation Algorithm, Register Allocation and Assignment.

Text Books

1 Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson Publishers,2007

Reference Books:

1.Compiler Construction, Principles and Practice, Kenneth C Louden,Cengage Learning, 2006
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kauffmann, 2001.
4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	TIME SERIES ANALYSIS (Program Elective-II)					

Course Objectives: The main objectives of introducing this course are

1. To introduce variety of statistical models for time series and cover main methods for analyzing these models. Understand the fundamental concepts of time series data, including its structures, graphical visualization, and basic modelling techniques.
2. Apply classical and advanced time series models such as AR, MA, ARMA, ARIMA, and seasonal models for effective forecasting of real-world data.
3. Evaluate and select appropriate forecasting models using statistical criteria, performance metrics, and modern approaches like neural networks and Bayesian methods.

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

CO1	Describe the characteristics of different types of time series data and perform graphical and numerical analyses for initial exploration.
CO2	Construct and fit time series models including AR, MA, ARMA, ARIMA, and seasonal ARIMA to forecast univariate time series data.
CO3	Analyze stationarity and invertibility of models, conduct model diagnostics, and perform variable selection for model refinement.
CO4	Apply multivariate time series models such as VAR and vector ARIMA, and utilize advanced forecasting techniques including neural networks and Bayesian methods
CO5	Solve practical forecasting problems by evaluating and selecting the most suitable model based on performance criteria

SYLLABUS	
UNIT-I:	**INTRODUCTION OF TIMESERIES ANALYSIS:**
Time Series and Forecasting, Different types of data, Internal structures of time series. Models for time series analysis, Autocorrelation and Partial autocorrelation. Examples of Time series Nature and uses of forecasting, Forecasting Process, Data for forecasting, Resources for forecasting	
UNIT-II:	**STATISTICS BACKGROUND FOR FORECASTING:**
Graphical Displays, Time Series Plots, Plotting Smoothed Data, Numerical Description of Time Series Data, Use of Data Transformations and Adjustments, General Approach to Time Series Modeling and Forecasting, Evaluating and Monitoring Forecasting Model Performance.	

UNIT-III:	TIME SERIES REGRESSION MODEL
Introduction Least Squares Estimation in Linear Regression Models, Statistical Inference in Linear Regression, Prediction of New Observations, Model Adequacy Checking, Variable Selection Methods in Regression, Generalized and Weighted Least Squares, Regression Models for General Time Series Data, Exponential Smoothing, First order and Second order.	
UNIT – IV	**AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA) MODELS**
Autoregressive Moving Average (ARMA) Models, Stationarity and Invertibility of ARMA Models, Checking for Stationarity using Variogram, Detecting Nonstationarity, Autoregressive Integrated Moving Average (ARIMA) Models, Forecasting using ARIMA, Seasonal Data, Seasonal ARIMA Models Forecasting using Seasonal ARIMA Models Introduction, Finding the "BEST" Model. **Example:** Internet Users Data Model Selection Criteria, Impulse Response Function to Study the Differences in Models Comparing Impulse Response Functions for Competing Models.	
UNIT-V	**MULTIVARIATE TIME SERIES MODELS AND FORECASTING:**
Multivariate Time Series Models and Forecasting, Multivariate Stationary Process, Vector ARIMA Models, Vector AR (VAR) Models, Neural Networks and Forecasting Spectral Analysis, Bayesian Methods in Forecasting.	
Text Books 1 Introduction To Time Series Analysis And Forecasting, 2nd Edition, Wiley Series In Probability And Statistics, By Douglas C. Montgomery, Cheryl L. Jen(2015) 2 Master Time Series Data Processing, Visualization, And Modeling Using Python Dr. Avishek PalDr. PksPrakash (2017)	

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	HIGH PERFORMANCE COMPUTING (Program Elective II)					

Course Objectives: The main objectives of introducing this course are

1. **Understand** the scope, motivation, and hardware platforms for parallel computing, including multi-core and GPU-based architectures.
2. **Apply** principles of parallel algorithm design, task decomposition, and communication techniques to solve computational problems efficiently.
3. **Develop and implement** parallel programs using shared memory models (OpenMP) and GPU programming frameworks (CUDA), focusing on performance optimization and scalability.

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

CO1	Demonstrate different parallel architectures, inter-connect networks, programming models
CO2	Apply decomposition, task mapping, and communication strategies to design parallel algorithms for problems such as matrix operations, sorting, and graph traversal.
CO3	Analyze and measure performance of modern parallel computing systems
CO4	Apply OpenMP and CUDA C to implement parallel applications for shared memory and GPU-based systems
CO5	Solve performance challenges in parallel programming by optimizing synchronization, memory usage, and communication in CUDA and OpenMP environments

SYLLABUS	
UNIT-I:	**INTRODUCTION**
Motivating Parallelism, Scope of Parallel Computing, Parallel Programming Platforms: Implicit Parallelism, Trends in Microprocessor and Architectures, Limitations of Memory, System performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines, Scalable design principles, Architectures: N-wide superscalar architectures, Multi- core architecture.	
UNIT-II:	**Parallel Programming:**
Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models, The Age of Parallel Processing, the Rise of GPU Computing, A Brief History of GPUs, Early GPU	

UNIT-III:	**Basic Communication**:
Operations- One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations. Programming shared address space platforms: threads- basics, synchronization, OpenMP programming	
UNIT – 4	**Analytical Models**
Sources of overhead in Parallel Programs, Performance Metrics for Parallel Systems, and The effect of Granularity on Performance, Scalability of Parallel Systems, Minimum execution time and minimum cost, optimal execution time. Dense Matrix Algorithms: Matrix Vector Multiplication, Matrix-Matrix Multiplication.	
UNIT-V:	**Parallel Algorithms- Sorting and Graph**
Issues in Sorting on Parallel Computers, Bubble Sort and its Variants, Parallelizing Quick sort, All- Pairs Shortest Paths, Algorithm for sparse graph, Parallel Depth-First Search, Parallel Best First Search. CUDA Architecture: CUDA Architecture, Using the CUDA Architecture, Applications of CUDA Introduction to CUDA C-Write and launch CUDA C kernels, Manage GPU memory, Manage communication and synchronization, Parallel programming in CUDA- C.	

Text Books

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel Computing", 2nd edition, Addison-Wesley, 2003, ISBN: 0-201-64865-2
2. Jason sanders, Edward Kandrot, “CUDA by Example”, Addison-Wesley, ISBN-13:978-0-13-138768-3

Reference Books

1. Kai Hwang, ”Scalable Parallel Computing”, McGraw Hill 1998, ISBN:0070317984

2. Shane Cook, “CUDA Programming: A Developer's Guide to Parallel Computing with GPUs”, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 2013 ISBN: 780124159884

3. David Culler Jaswinder Pal Singh, ”Parallel Computer Architecture: A Hardware/ Software Approach”, Morgan Kaufmann,1999, ISBN 978-1-55860-343-1

4. Rod Stephens, “Essential Algorithms”, Wiley, ISBN: ISBN: 978-1-118-61210-1

M Tech I Semester		PE	L	T	P	C
			3	0	0	3
	AGILE METHODOLOGIES (Program Elective II)					

Course Objectives: The main objectives of introducing this course are

1. **Understand** the values, principles, and mindset behind Agile methodologies, including Scrum, XP, Lean, and Kanban, and their application in modern software development.
2. **Apply** Agile practices such as user stories, sprints, retrospectives, continuous integration, and WIP limits to plan, manage, and deliver high-value software projects effectively.
3. **Evaluate and reflect** on team dynamics, communication, coaching principles, and organizational culture to foster continuous improvement and agility within teams and projects.

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

CO1	Explain the core values and principles of the Agile Manifesto and differentiate between Agile methodologies.
CO2	Apply Agile practices such as writing user stories, managing sprints, conducting retrospectives
CO3	Analyze team behaviors, communication breakdowns, and the impact of organizational culture on Agile adoption and project success.
CO4	Evaluate Agile project performance by using metrics
CO5	Design and Implement an Agile process tailored to a specific project or team context, incorporating various practices.

SYLLABUS	
UNIT-I:	**Introduction: Learning Agile**
Getting Agile into your brain, Understanding Agile values, No Silver Bullet, Agile to the Rescue, adding Agile makes a difference. A fractured perspective, How a fractured perspective causes project problems. The Agile Manifesto, Purpose behind Each Practice. Individuals and Interactions Over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract Negotiation, Responding to Change over Following a Plan, Principles over Practices. Understanding the Elephant, Methodologies Help You Get It All in Place at Once, Where to Start with a New Methodology	

UNIT-II:	**The Agile Principles:**
The 12 Principles of Agile Software, The Customer Is Always Right, "Do As I Say, Not As I Said". Delivering the Project, BetterProject Delivery for the Ebook Reader Project. Communicating and Working Together, Better Communication for the Ebook Reader Project. Project Execution—Moving the Project Along, A Better Working Environment for the Ebook Reader Project Team. Constantly Improving the Project and the Team. The Agile Project: Bringing All the Principles Together	
UNIT-III:	**SCRUM and Self-Organizing Teams:**
The Rules of Scrum, Act I: I Can Haz Scrum?, Everyone on a Scrum Team owns the Project, The Scrum Master Guides the Team's Decisions, The Product Owner Helps the Team Understand the Value of the Software, Everyone Owns the Project, Scrum Has Its Own Set of Values Status Updates Are for Social Networks!, The Whole Team Uses the Daily Scrum, Feedback and the Visibility- Inspection-Adaptation Cycle, The Last Responsible Moment, How to Hold an Effective Daily Scrum. Sprinting into a Wall, Sprints, Planning, and Retrospectives, Iterative or Incremental?, The Product Owner Makes or Breaks the Sprint, Visibility and Value, How to Plan and Run an Effective Scrum Sprint Scrum Planning And Collective Commitment: Not Quite Expecting the Unexpected, User Stories, Velocity, and Generally Accepted Scrum Practices, Make Your Software Useful, User Stories Help Build Features Your Users Will Use, Conditions of Satisfaction, Story Points and Velocity, Burndown Charts, Planning and Running a Sprint Using Stories, Points, Tasks, and a Task Board. Victory Lap, Scrum Values Revisited, Practices Do Work Without the Values (Just Don't Call It Scrum), Is Your Company's Culture Compatible with Scrum Values.	
UNIT – IV	**XP And Embracing Change:**
Going into Overtime, The Primary Practices of XP, Programming Practices, Integration Practices, Planning Practices, Team Practices, Why Teams Resist Changes, and How the Practices Help. The Game Plan Changed, but We're Still Losing, The XP Values Help the Team Change Their Mindset, XP Helps Developers Learn to Work with Users, Practices Only "Stick" When the Team Truly Believes in Them, An Effective Mindset Starts with the XP Values, The XP Values, Paved with Good Intentions. The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change, The Principles of XP, XP Principles Help You Understand Planning, XP Principles Help You Understand Practices—and Vice Versa, Feedback Loops. XP, Simplicity, and Incremental Design: Code and Design, Code Smells and Antipatterns (or, How to Tell If You're Being Too Clever), XP Teams Look for Code Smells and Fix Them, Hooks, Edge Cases, and Code That Does Too Much. Make Code and Design Decisions at the Last Responsible Moment, Fix Technical Debt	

by Refactoring Mercilessly, Use Continuous Integration to Find Design Problems, Avoid Monolithic Design, Incremental Design and the Holistic XP Practices. Teams Work Best When They Feel Like They Have Time to Think, Team Members Trust Each Other and Make Decisions Together. The XP Design Planning, Team, and Holistic Practices Form an Ecosystem Incremental Design Versus Designing for Reuse, When Units Interact in a Simple Way, the System Can Grow Incrementally,
Great Design Emerges from Simple Interactions, Final Score.

UNIT-V:	**Lean, Eliminating Waste, and Seeing the whole:**

Lean Thinking, Commitment, Options Thinking, and Set-Based Development, Creating Heroes and Magical Thinking. Eliminate Waste, Use a Value Stream Map to Help See Waste Clearly, Gain a Deeper Understanding of the Product, See the Whole, Find the Root Cause of Problems That You Discover. Deliver As Fast As Possible, Use an Area Chart to Visualize Work in Progress, Control Bottlenecks by Limiting Work in Progress. Kanban, Flow, and Constantly Improving: The Principles of Kanban, Find a Starting Point and Evolve Experimentally from There. Stories Go into the System; Code Comes Out, Improving Your Process with Kanban, Visualize the Workflow, Limit Work in Progress. Measure and Manage Flow, Managing Flow with WIP Limits Naturally Creates Slack. Make Process Policies Explicit So Everyone Is on the Same Page. Emergent Behavior with Kanban.
The Agile Coach: Coaches Understand Why People Don't Always Want to Change. The Principles of Coaching

Text Books

1 Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

Reference Books

1. Andrew stellman, Jennifer Green, Head first Agile, O'Reilly, 2017.
2. Rubin K , Essential Scrum : A practical guide to the most popular Agile process, Addison-Wesley, 2013

M Tech I Semester			L	T	P	C
			0	1	2	2
	DATA STRUCTURES AND ALGORITHM ANALYSIS LAB					

Course Objectives: The main objectives of introducing this course are

1. To enable students to analyze and evaluate the correctness and efficiency of algorithms using appropriate mathematical techniques.
2. To provide students with the ability to understand, implement and apply a wide range of abstract data types (ADTs) and advanced data structures in solving real-world problems.
3. To equip students with the knowledge to design, compare and optimize search trees, hashing methods, and sorting techniques for various applications in computer science.

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

CO1	Analyze and determine the correctness and time-space complexity of algorithms using asymptotic notation.
CO2	Develop and implement advanced abstract data types (ADTs) and data structures to solve complex computational problems
CO3	Demonstrate various searching, sorting and hash techniques and be able to apply and solve problems of real life
CO4	Apply data structures including linked lists, binary trees, heaps, graphs, and search trees to design and implement computational solutions
CO5	Build search tree solutions to solve IT-related problems.

List of Experiments	
Experiment-1	Write a java program to perform various operations on single linked List
Experiment-2	Write a java program for the following a) Reverse a linked list b) Sort the data in a linked list c) Remove duplicates d) Merge two linked lists
Experiment-3	Write a java program to perform various operations on doubly linked List
Experiment-4	Write a java program to perform various operations on circular linked List
Experiment-5	Write a java program for performing various operations on stack using linked list
Experiment-6	Write a java program for performing various operations on queue using linked list

Experiment-7	Write a java program for the following using stack a) Infix to postfix conversion. b) Expression evaluation. c) Obtain the binary number for a given decimal number.
Experiment-8	Write a java program to implement various operations on Binary Search Tree Using Recursive and Non-Recursive methods.
Experiment-9	Write a java program to implement the following for a graph. a) BFS b) DFS
Experiment-10	Write a java program to implement Merge & Heap Sort of given Elements
Experiment-11	Write a java program to implement Quick Sort of given elements
Experiment-12	Write a java program to implement various operations on AVL trees
Experiment-13	Write a java program to perform the following operations: a) Insertion into a B-tree b) Searching in a B-tree
Experiment-14	Write a java program to implementation of recursive and non- recursive functions to Binary tree Traversals
Experiment-15	Write a java program to implement all the functions of Dictionary (ADT) using Hashing

M Tech I Semester			L	T	P	C
			0	1	2	2
	ADVANCED DATA WAREHOUSING AND DATA MINING LAB					

Course Objectives: The main objectives of introducing this course are

1. **To understand** the fundamental concepts and architectures of data warehouses and OLAP systems, including their design, modeling, and real-world deployment.
2. **To develop the ability** to apply data mining techniques using tools like WEKA and Python for analyzing large real-world datasets and extracting meaningful insights.
3. **To enable students** to design, implement, and evaluate custom data mining algorithms for solving practical business and technical problems. Develop ability to design various algorithms based on data mining tools.

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

CO1	Build data warehouses and data marts using industry-standard tools, and perform OLAP operations such as roll-up, drill-down, slice, and dice.
CO2	Analyze datasets in **WEKA** and **Python/R** by performing preprocessing, association rule mining, and classification using algorithms like Apriori, ID3, J48, Naïve Bayes, and k-NN.
CO3	Compare and evaluate the performance of various classification and clustering algorithms using metrics such as confusion matrix, ROC curves, entropy, Kappa statistic, and SSE.
CO4	Develop and Implement custom programs in **Python, Java, or R** to perform key data mining tasks such as association rule mining, clustering, and statistical computations (e.g., chi-square).
CO5	Utilize graphical techniques and tools like Matplotlib and R to visualize and interpret complex data for insights into distributions and patterns

List of Experiments	
Experiment-1	Creation of a Data Warehouse. • Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects,etc.,) • Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc). • Write ETL scripts and implement using data warehouse tools. • Perform Various OLAP operations such slice, dice, roll up, drill up and pivot

Experiment-2	Explore machine learning tool "WEKA" • Explore WEKA Data Mining/Machine Learning Toolkit. • Downloading and/or installation of WEKA data mining toolkit. • Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface. • Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel) • Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather dataset, Iris dataset, etc.) • Load each dataset and observe the following: 1. List the attribute names and they types 2. Number of records in each dataset 3. Identify the class attribute (if any) 4. Plot Histogram 5. Determine the number of records for each class. 6. Visualize the data in various dimensions
Experiment-3	Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets 1. Explore various options available in Weka for preprocessing data and apply 2. Unsupervised filters like Discretization, Resample filter, etc. on each dataset 3. Load weather. nominal, Iris, Glass datasets into Weka and run Apriori 4. Algorithm with different support and confidence values. 5. Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. 6. Derive interesting insights and observe the effect of discretization in the rule generation process
Experiment-4	Demonstrate performing classification on data sets Weka/R • Load each dataset and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic. • Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.

	• Load each dataset into Weka/R and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained. • Plot RoC Curves • Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.
Experiment-5	Demonstrate performing clustering of data sets • Load each dataset into Weka/R and run simple k-means clustering algorithm with different values of k (number of desired clusters). • Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights. • Explore other clustering techniques available in Weka/R. • Explore visualization features of Weka/R to visualize the clusters. Derive interesting insights and explain
Experiment-6	Demonstrate knowledge flow application on data sets into Weka/R • Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms • Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree
Experiment-7	Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations
Experiment-8	Write a java program to prepare a simulated data set with unique instances
Experiment-9	Write a Python program to generate frequent item sets / association rules using Apriori algorithm
Experiment-10	Write a program to calculate chi-square value using Python/R. Report your observation
Experiment-11	Implement a Java/R program to perform Apriori algorithm
Experiment-12	Write a R program to cluster your choice of data using simple k-means algorithm using JDK
Experiment-13	Write a program of cluster analysis using simple k-means algorithm Python/R programming language

Experiment-14	Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python
Experiment-15	Visualize the datasets using matplotlib in python/R.(Histogram, Box plot, Bar chart, Pie chart etc.,)

M Tech II Semester			L	T	P	C
			3	1	0	4
	MACHINE LEARNING					

Course Objectives: The main objectives of introducing this course are

1. Define machine learning and its different types (supervised and unsupervised) and understand their applications
2. Apply supervised learning algorithms including decision trees and k-nearest neighbours – KNN
3. Implement unsupervised learning techniques, such as K-means clustering

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

CO1	Explain the basic concepts, paradigms, and processes involved in machine learning.
CO2	Apply nearest neighbor-based models and distance measures for classification and regression tasks.
CO3	Build and evaluate decision tree, random forest, and Bayesian models for classification and regression.
CO4	Apply linear discriminants, SVMs, logistic regression, and neural networks for predictive modeling.
CO5	Analyze and apply clustering algorithms for partitioning and pattern discovery in datasets.

SYLLABUS	
UNIT-I:	**Introduction to Machine Learning:**
Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and Learning, Data Sets.	
UNIT-II:	**Nearest Neighbor-Based Models:**
Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms.	
UNIT-III:	**Models Based on Decision Trees**
Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression.	

The Bayes Classifier: Introduction to the Bayes Classifier, Bayes' Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification | Class Conditional Independence and Naive Bayes Classifier (NBC)	
UNIT – IV	**Linear Discriminants for Machine Learning**
Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptron's (MLPs), Backpropagation for Training an MLP	
UNIT-V:	**Clustering**
Introduction to Clustering, Partitioning of Data, Matrix Factorization | Clustering of Patterns, Divisive Clustering, Agglomerative Clustering, Partitional Clustering, K-Means Clustering, Soft Partitioning, Soft Clustering, Fuzzy C-Means Clustering, Rough Clustering, Rough K- Means Clustering Algorithm, Expectation Maximization-Based Clustering, Spectral Clustering	
Text Books 1 "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024	

Reference Books:

1. Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017
2. "Machine Learning in Action",Peter Harrington, DreamTech
3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019.

M Tech II Semester		PC	L	T	P	C
			3	1	0	4
	CRYPTOGRAPHY & NETWORK SECURITY					

Course Objectives: The main objectives of introducing this course are

1. Explain the objectives of information security.
2. Explain the importance and application of each of confidentiality, integrity, authentication and availability.
3. Understand the basic categories of threats to computers and networks
4. Discuss the Mathematics of Cryptography.
5. Discuss various layers protocols enhanced security mechanisms.

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

CO1	Demonstrate security issues related to computer networks and learn different symmetric key techniques
CO2	Apply mathematic of cryptography for symmetric and Asymmetric algorithms and apply this knowledge to understand the Cryptographic algorithms
CO3	Compare and contrast different types of symmetric and Asymmetric algorithms
CO4	Explain Hash functions, message authentication and digital signature and their importance to the security
CO5	Analyze enhanced security protocols at various layers.

SYLLABUS	
UNIT-I:	**Security Concepts**
Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security Cryptography. Classical Encryption Techniques-symmetric cipher model, Substitution techniques, Transposition techniques, Rotor Machines, Stegnography.	
UNIT-II:	**Introduction to Symmetric Cryptography**
Algebraic Structures-Groups, Rings, Fields, GF fields, Polynomials. Mathematics of Asymmetric cryptography: Primes, Checking For Primness, Eulers phi-functions, Fermat's Little Theorem, Euler's Theorem, Generating Primes, Primality Testing, Factorization, Chinese Remainder Theorem, Quadratic Congruence, Exponentiation And Logarithm.	

UNIT-III:	**Symmetric key Ciphers:**

Block Cipher principles, DES, AES, Blowfish, IDEA, Block cipher operation, Stream ciphers: RC4, RC5 Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Diffie- Hellman Key Exchange, Elgamal Cryptographic system, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

UNIT – IV	**Cryptographic Hash Functions:**

Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithms (SHA) Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MAC'S,MAC'S Based On Hash Functions: HMAC, MAC'S Based On Block Ciphers: DAA And CMAC

Digital Signatures: Digital Signatures, Elgamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm.

UNIT-V:	**Network and Internet Security: Transport-Level Security:**

Web Security Considerations, Transport Level Security, HTTPS, SSH. IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Authentication Header Protocol. Electronic-Mail Security: Internet-mail Security, Email Format, Email Threats and Comprehensive Email Security, S/MIME, PGP.

Text Books

1. Cryptography and Network Security - Principles and Practice: William Stallings, Pearson Education, 7th Edition, 2017
2. Cryptography and Network Security: Behrouz A. Forouzan Debdeep, Mc Graw Hill, 3rd Edition, 2015

REFERENCE BOOKS:

1. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition
2. Introduction to Cryptography with Coding Theory: Wade Trappe, Lawrence C. Washington, Pearson.
3. Modern Cryptography: Theory and Practice By Wenbo Mao. Pearson

M Tech II Semester			L	T	P	C
			3	1	0	4
	INTRODUCTION TO QUANTUM COMPUTING					

Course Objectives: The main objectives of introducing this course are

1. Introduce fundamental concepts of quantum mechanics and its mathematical formalism.
2. Explore quantum computing and communication principles and technologies.
3. Understand the physical implementation and limitations of quantum systems
4. Enable students to relate quantum theory to practical applications in computing, cryptography and sensing.
5. Familiarize students with the emerging trends in quantum technologies

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

CO1	Explain the history and foundational concepts of quantum computing including qubits and quantum logic.
CO2	Apply the principles of linear algebra, quantum mechanics, and biology relevant to quantum computing.
CO3	Demonstrate the representation of qubits, Bloch sphere, and design of basic quantum circuits.
CO4	Analyze and apply fundamental quantum algorithms such as Deutsch, Shor, and Grover.
CO5	Apply noise and error correction methods to evaluate and implement cryptographic applications in quantum computing

SYLLABUS	
UNIT-I:	**History of Quantum Computing:**
Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations	
UNIT-II:	**Background Mathematics:**
Basics of Linear Algebra, Hilbert space, Probabilities and measurements. Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)	
UNIT-III:	**Qubit:**
Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.	

UNIT – IV	**Quantum Algorithms:**
Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm	
UNIT-V:	**Noise and error correction:**
Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation	

Text Books

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge

Reference Books:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol.I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms

M Tech II Semester		PE	L	T	P	C
			3	0	0	3
	FEATURE ENGINEERING (Program Elective- III)					

Course Objectives: The main objectives of introducing this course are

1. Describe the Basic concepts of Data, Tasks, Models, Features and Model building.
2. Explain the concept of converting Text into Flat Vectors using Bag- of- Words, and Bag- of- n-Grams.
3. Demonstrate techniques for Dimensionality Reduction.
4. Discuss on Non linear Featurization.
5. Explain the concept of Item- Based Collaborative Filtering.

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

CO1	Apply data preprocessing and feature transformation techniques for effective model building.
CO2	Utilize Bag-of-Words, Tf-Idf, and classification models to process and represent text data
CO3	Develop encoding techniques for categorical variables and dimensionality reduction methods using PCA..
CO4	Utilize k-means clustering for nonlinear featurization and model stacking.
CO5	Develop and improve recommender systems using item-based collaborative filtering approaches.

SYLLABUS	
UNIT-I:	**The Machine Learning Pipeline**
Data, Tasks, Models, Features, Model Evaluation Fancy Tricks with Simple Numbers: Scalars, Vectors, and Spaces, Dealing with Counts, Binarization, Quantization or Binning, Log Transformation, Log Transform in Action, Power Transforms: Generalization of the Log Transform, Feature Scaling or Normalization, Min-Max Scaling, Standardization (Variance Scaling), ℓ2 Normalization, Interaction Features, Feature Selection	
UNIT-II:	**Text Data: Flattening, Filtering, and Chunking: Bag-of-X:**
Turning Natural Text into Flat Vectors, Bag- of-Words, Bag-of-n-Grams, Filtering for Cleaner Features: Stopwords, Frequency-Based Filtering, Stemming; Atoms of Meaning: From Words to n- Grams to Phrases: Parsing and Tokenization, Collocation Extraction for Phrase Detection The Effects of Feature Scaling: From Bag-of-Words to Tf-Idf :Tf-Idf : A Simple Twist on Bag-of- Words, Putting	

It to the Test : Creating a Classification Dataset, Scaling Bag-of-Words with Tf-Idf Transformation, Classification with Logistic Regression, Tuning Logistic Regression with Regularization.	
UNIT-III:	**Categorical Variables:**
Counting Eggs in the Age of Robotic Chickens: Encoding Categorical Variables: One-Hot Encoding, Dummy Coding, Effect Coding, Pros and Cons of Categorical Variable Encodings; Dealing with Large Categorical Variables: Feature Hashing, Bin Counting. Dimensionality Reduction: Squashing the Data Pancake with PCA: Intuition, Derivation: Linear Projection, Variance and Empirical Variance, Principal Components: First Formulation, Principal Components: Matrix- Vector Formulation, General Solution of the Principal Components; Transforming Features, Implementing PCA: PCA in Action, Whitening and ZCA, Considerations and Limitations of PCA	
UNIT – IV	**Nonlinear Featurization via K-Means Model Stacking:**
k-Means Clustering, Clustering as Surface Tiling, k-Means Featurization for Classification: Alternative Dense Featurization, Pros, Cons, and Gotchas	
UNIT-V:	**Item-Based Collaborative Filtering, First Pass:**
Data Import, Cleaning, and Feature Parsing, Academic Paper Recommender: Naive Approach, Second Pass: More Engineering and a Smarter Model, Academic Paper Recommender: Take 2, Third Pass: More Features is More Information, Academic Paper Recommender: Take 3	
Text Books 1. "Feature Engineering for Machine Learning Principles and Techniques for Data Scientists", Alice Zheng& Amanda Casari, O'REILLY, 2018 2. "Feature Engineering and Selection: A Practical Approach for Predictive Models", Max Kuhn, Kjell Johnson, CRC Press, 2019	

M Tech II Semester		PE	L	T	P	C
			3	0	0	3
	GENERATIVE AI (Program Elective- III)					

Course Objectives: The main objectives of introducing this course are

1. To learn fundamental skills for Generative AI
2. To study techniques for cleaning and preparing data for Generative AI tasks
3. To implement Generative AI models
4. To develop innovative applications using generative AI tools and techniques.

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

CO1	Explain the foundations, types, challenges, and ethical aspects of generative AI.
CO2	Apply language models, transformer architectures, and prompt engineering techniques for text generation.
CO3	Utilize GANs, VAEs, diffusion, and transformer-based models for image generation
CO4	Demonstrate the use of GAN variants, style transfer, and neural networks for generating paintings, music, and plays.
CO5	Develop and fine-tune generative AI models using open-source frameworks and deployment tools.

SYLLABUS	
UNIT-I:	**Introduction To Gen Ai:**
Historical Overview of Generative modeling, Difference between Gen AI and Discriminative Modeling, Importance of generative models in AI and Machine Learning, Types of Generative models, GANs, VAEs, autoregressive models and Vector quantized Diffusion models, Understanding if probabilistic modeling and generative process, Challenges of Generative Modeling, Future of Gen AI, Ethical Aspects of AI, Responsible AI, Use Cases	
UNIT-II:	**Generative Models For Text:**
Language Models Basics, Building blocks of Language models, Transformer Architecture, Encoder and Decoder, Attention mechanisms, Generation of Text, Models like BERT and GPT models, Generation of Text, Auto encoding, Regression Models, Exploring Catgut, Prompt Engineering: Designing Prompts, Revising Prompts using Reinforcement Learning from Human Feedback (RLHF), Retrieval Augmented Generation, Multimodal LLM, Issues of LLM like hallucination	

UNIT-III:	Generation of Images:
. Introduction to Generative Adversarial Networks, Adversarial Training Process, Nash Equilibrium, Variation Auto encoders, Encoder-Decoder Architectures, Stable Diffusion Models, Introduction to Transformer-based Image Generation, CLIP, Visual Transformers Vat- Dall-E2 and Dall-E3, GPT-4V, Issues of Image Generation models like Mode Collapse and Stability.	
UNIT – IV	**Generation of Painting, Music, and Play:**
Variants of GAN, Types of GAN, Cyclic GAN, Using Cyclic GAN to Generate Paintings, Neural Style Transfer, Style Transfer, Music Generating RNN, Muskegon, Autonomous agents, Deep Q Algorithm, Actor-critic Network.	
UNIT-V:	**Open Source Models And Programming Frameworks:**
Training and Fine tuning of Generative models, GPT 4 All, Transfer learning and Retrained models, Training vision models, Google Copilot, Programming LLM, Lang Chain, Open Source Models, Llama, Programming for Times former, Deployment, Hugging Face.	

Text Books

1. Denis Rothman, “Transformers for Natural Language Processing and Computer Vision”, Third Edition , Packet Books, 2024

Reference Books:

1. David Foster,”Generative Deep Learning”, O’Neil Books, 2024.
2. Altar Remain, “Generative AI for Everyone”, Blue Rose One, 2024.

M Tech II Semester			L	T	P	C
			3	0	0	3
	DESIGN PATTERNS (Program Elective- III)					

Course Objectives: The main objectives of introducing this course are

1. Demonstration of patterns related to object oriented Design
2. Describe the design patterns that are common in software applications.
3. Analyze a software development problem and express it.
4. Design a module structure to solve a problem and evaluate alternatives
5. Implement a module so that it executes efficiently and correctly

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

CO1	Classify design patterns by purpose and apply them to solve object-oriented design problems.
CO2	Build object creation mechanisms by applying creational patterns like Singleton and Abstract Factory.
CO3	Apply structural patterns to compose classes and objects into larger structures.
CO4	Utilize behavioral patterns to manage effective communication and responsibility between objects.
CO5	Build algorithms and manage state by applying behavioral patterns like Strategy and State.

SYLLABUS	
UNIT-I:	**Introduction**
What is a Design Pattern, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalogue of Design Patterns, Organizing The Catalog, How Design Patterns solve Design Problems, How to Select a Design pattern, How to Use a Design Pattern.	
UNIT-II:	**A Case Study**
Designing a Document Editor, Design Problems ,Document Structure, Formatting , Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary, Creational Patterns, Abstract Factory, Builder , Factory Method, Prototype, Singleton, Discussion of Creational Patterns.	
UNIT-III:	**Structural Patterns**
Structural Pattern Part-I, Adapter, Bridge, Composite. Structural Pattern Part-II, Decorator, Facade, Flyweight, Proxy.	

UNIT – IV	**Behavioral Patterns**
Behavioral Patterns Part: I, Chain of Responsibility, Command, Interpreter, Iterator. Behavioral Patterns Part: II, Mediator, Memento, Observer, Discussion of Behavioral Patterns.	
UNIT-V:	**Behavioral Patterns Part:** III,
State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns. What to Expect from Design Patterns, A Brief History, The Pattern Community, An Invitation, A Parting Thought	
Text Books 1. Design Patterns By Erich Gamma, Pearson Education	

Reference Books:

1. Patterns in JAVA Vol-I (or) Vol-II By Mark Grand, Wiley Dream Tech.

2. Java Enterprise Design Patterns Vol-III By Mark Grand Wiley Dream Tech

M Tech II Semester			L	T	P	C
			3	0	0	3
	Natural Language Processing					

Course Objectives: The main objectives of introducing this course are

1. Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information
2. The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.
3. Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

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

CO1	Explain the origins, challenges, and basic processes of Natural Language Processing.
CO2	Apply n-gram models and part-of-speech tagging methods for word-level analysis.
CO3	Construct and analyze syntactic structures using grammar rules and parsing techniques.
CO4	Design a tag set to be used for statistical processing for real-time Applications
CO5	Utilize discourse analysis methods and lexical resources for advanced NLP applications.

SYLLABUS	
UNIT-I:	**INTRODUCTION**
Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance	
UNIT-II:	**WORD LEVEL ANALYSIS**
Unsmoothed N-grams, Evaluating N- grams, Smoothing, Interpolation and Backoff – Word Classes, Part- of- Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models	
UNIT-III:	**SYNTACTIC ANALYSIS**
Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures,	

Unification of feature structures

UNIT – IV	SEMANTICS AND PRAGMATICS

Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

UNIT-V:	DISCOURSE ANALYSIS AND LEXICAL RESOURCES

Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC)

Text Books

1 Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2nd Edition, Daniel Jurafsky, James H. Martin - Pearson Publication,2014.

2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.

Reference Books:

1. Language Processing with Java and Ling Pipe Cookbook, 1st Edition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2nd Edition, Richard M Reese, OReilly Media,2015.
3. Handbook of Natural Language Processing, Second, Nitin Indurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition
4. Natural Language Processing and Information Retrieval, 3rd Edition, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press,2008.

M Tech II Semester			L	T	P	C
			3	0	0	3
	BLOCK CHAIN TECHNOLOGIES (Program Elective- IV)					

Course Objectives: The main objectives of introducing this course are

1. Architect sensor networks for various application setups.
2. Devise appropriate data dissemination protocols and model links cost.
3. Understandings of the fundamental concepts of wireless sensor networks and have a basic knowledge of the various protocols at various layers.
4. Evaluate the performance of sensor networks and identify bottlenecks

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

CO1	Demonstrate Cryptographic primitives used in Blockchain
CO2	Develop a model of Bitcoin's structure to demonstrate solutions for double-spending and forks.
CO3	Compare PoW and PoS consensus mechanisms and their key properties.
CO4	Develop basic Smart Contracts and contrast Ethereum with Bitcoin scripting.
CO5	Apply blockchain principles to develop a functional application model and evaluate its limitations.

SYLLABUS	
UNIT-I:	**INTRODUCTION TO BLOCKCHAIN**
Introduction, history of Bitcoin and origins of Blockchain, Fundamentals of Blockchain and key components, Permission and Permission-less platforms, Introduction to Cryptography, SHA256 and ECDSA, Hashing and Encryption, Symmetric/ Asymmetric keys, Private and Public Keys.	
UNIT-II:	**TECHNOLOGIES BORROWED IN BLOCKCHAIN**
Technologies Borrowed in Blockchain –hash pointers- - Digital cash etc.- Bitcoin blockchain - Wallet – Blocks Merkley Tree - hardness of mining - Transaction verifiability - Anonymity - forks - Double spending - Mathematical analysis of properties of Bitcoin - Bitcoin- the challenges and solutions.	

UNIT-III:	CONSENSUS MECHANISMS
Consensus Algorithms: Proof of Work (PoW) as random oracle - Formal treatment of consistency- Liveness and Fairness - Proof of Stake (PoS) based Chains -Hybrid models (PoW + PoS), Byzantine Models of fault tolerance.	
UNIT – IV	**ETHEREUM**
Ethereum -Ethereum Virtual Machine (EVM) -Wallets for Ethereum -Solidity - Smart Contracts (Chapter 5-book1), - The Turing Completeness of Smart Contract Languages and verification challenges- Using smart contracts to enforce legal contracts- Comparing Bitcoin scripting vs. Ethereum Smart Contracts-Some attacks on smart contracts.	
UNIT-V:	**HYPERLEDGER FABRIC**
Hyperledger fabric- the plug and play platform and mechanisms in permissioned blockchain – Beyond Cryptocurrency – applications of blockchain in cyber security- integrity of information- E Governance and other contract enforcement mechanisms - Limitations of blockchain as a technology and myths vs reality of blockchain technology.	

Text Books

1. Blockchain Technology Chandramouli Subramanian, Asha A George, Abhilash K A and Meena Karthikeyan, University Press, 2020.
2. Mastering Blockchain - Distributed ledger technology, decentralization, and smart contracts explained, Imran Bashir,2nd ed. Edition,2018, pakct publication

Reference Books:

1. .Shukla, M.Dhawan, S.Sharma,S. Venkatesan "Blockchain Technology: Cryptocurrency and Applications" ,Oxford University Press 2019 .

2. Cryptography and network security principles and practice, William Stallings, Pearson, 8th edition,

WEB REFERENCES:

1. https://drive.google.com/file/d/1PtYaDmWYaqPVGjKDnMYGWO5eoI5wMPtJ/view
2. https://archive.nptel.ac.in/courses/106/104/106104220/
3. https://www.tutorialspoint.com/blockchain/index.htm

M Tech II Semester			L	T	P	C
			3	0	0	3
	ADHOC SENSOR NETWORKS (Program Elective- IV)					

Course Objectives: The main objectives of introducing this course are

1. Architect sensor networks for various application setups.
2. Devise appropriate data dissemination protocols and model links cost.
3. Understandings of the fundamental concepts of wireless sensor networks and have a basic knowledge of the various protocols at various layers.
4. Evaluate the performance of Sensor Networks and identify bottlenecks.

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

CO1	Explain the fundamentals, architectures, and design challenges of ad hoc and sensor networks.
CO2	Analyze MAC protocol issues, goals, and classifications for ad hoc wireless networks.
CO3	Develop routing and transport layer protocols with emphasis on security in ad hoc wireless networks.
CO4	Demonstrate sensor node architecture, network organization, and MAC protocols in WSN.
CO5	Apply routing, localization, synchronization, and QoS strategies for efficient WSN design.

SYLLABUS	
UNIT-I:	**Introduction:**
Fundamentals of Wireless Communication Technology, The Electromagnetic Spectrum, Radio propagation Mechanisms, Characteristics of the Wireless channel mobile ad hoc networks (MANETs), Wireless Sensor Networks (WSNs): concepts and architectures, Applications of Ad Hoc and Sensor Networks, Design Challenges in Ad hoc and Sensor Networks	
UNIT-II:	**MAC Protocols for Ad Hoc Wireless Networks:**
Issues in designing a MAC Protocol, Issues in Designing a MAC Protocol for Ad Hoc Wireless Networks, Design Goals of a MAC Protocol for Ad Hoc Wireless Networks, Classification of MAC Protocols, Contention based protocols, Contention based protocols with Reservation Mechanisms, Contention based protocols with Scheduling Mechanisms, Multi-channel MAC – IEEE 802.11	

UNIT-III:	**Routing Protocols and Transport Layer In Ad Hoc Wireless Networks:**
Routing Protocol: Issues in designing a routing protocol for Adhoc networks, Classification, proactive routing, reactive routing (on- demand), hybrid routing, Transport Layer protocol for Ad hoc networks, Design Goals of a Transport Layer Protocol for AdHoc Wireless Networks, Classification of Transport Layer solutions-TCP over Ad hoc wireless, Network Security, Security in Ad Hoc Wireless Networks, Network Security Requirements	
UNIT – IV	**Wireless Sensor Networks (WSNS) And Mac Protocols:**
Single node architecture - hardware and software components of a sensor node, WSN Network architecture: typical network architectures, data relaying and aggregation strategies, MAC layer protocols: self-organizing, Hybrid TDMA/FDMA and CSMA based MAC -IEEE 802.15.4.	
UNIT-V:	**WSN Routing, Localization &QoS:**
Issues in WSN routing, OLSR, Localization, Indoor and Sensor Network Localization, absolute and relative localization, triangulation, QOS in WSN, Energy Efficient Design, Synchronization.	

Text Books

1. Ad Hoc Wireless Networks: Architectures and Protocols, C. Siva Ram Murthy, and B. S. Manoj, Pearson Education, 2008
2. "Wireless Adhoc and Sensor Networks", Labiod. H, Wiley, 1 st edition-2008
3. "Wireless ad -hoc and sensor Networks: theory and applications", Li, X, Cambridge University Press, fifth edition-2008.

Reference Books:

1. "Ad Hoc & Sensor Networks: Theory and Applications", 2nd edition, Carlos De MoraisCordeiro, Dharma Prakash Agrawal ,World Scientific Publishing Company, 2011
2. Wireless Sensor Networks Feng Zhao and LeonidesGuibas,Elsevier Publication 2nd edition-2004
3. "Protocols and Architectures for Wireless Sensor Networks", Holger Karl and Andreas Willig,Wiley, 2005 .
4. "Wireless Sensor Networks Technology, Protocols, and Applications", KazemSohraby, Daniel Minoli, &TaiebZnati, John Wiley, 2007.

M Tech II Semester			L	T	P	C
			3	0	0	3
	DEVOPS **(Program Elective- III)**					

Course Objectives: The main objectives of introducing this course are

1. Describe the agile relationship between development and IT operations.
2. Understand the skill sets and high-functioning teams involved in DevOps and related methods to reach a continuous delivery capability.
3. Implement automated system update and DevOps lifecycle.

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

CO1	Apply DevOps principles and architecture to design an automated CI/CD pipeline for a given Agile workflow
CO2	Apply GIT workflows and branching strategies to manage collaborative code development and integrate unit testing for quality analysis.
CO3	Develop and manage a Jenkins pipeline to automate the build and integration process within a master-slave architecture.
CO4	Construct a continuous delivery pipeline by building Docker containers to package and deploy applications.
CO5	Build automated deployment scripts using Ansible and construct containerized solutions on Kubernetes.

SYLLABUS	
UNIT-I:	**Introduction to DevOps:**
Introduction to SDLC, Agile Model. Introduction to DevOps. DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/ CD. Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples	
UNIT-II:	**Source Code Management (GIT):**
The need for source code control, The history of source code management, Roles and code, source code management system and migrations. What is Version Control and GIT, GIT Installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration. UNIT TESTING-CODECOVERAGE: Junit ,nUnit & Code Coverage with Sonar Qube, SonarQube - Code Quality Analysis.	

UNIT-III:	**Build Automation - Continuous Integration (CI):**

Build Automation, What is CI Why Cl is Required, CI tools, Introduction to Jenkins (With Architecture), jenkins workflow, jenkins master slave architecture, Jenkins Pipelines, PIPELINE BASICS - Jenkins Master, Node, Agent, and Executor Freestyle Projects& Pipelines, Jenkins for Continuous Integration, Create and Manage Builds, User Management in Jenkins Schedule Builds,
Launch Builds on Slave Nodes.

UNIT – IV	**Continuous Delivery:**

Importance of Continuous Delivery, CONTINUOUS DEPLOYMENT CD Flow, Containerization with Docker: Introduction to Docker, Docker installation, Docker commands, Images & Containers, Docker File, running containers, working with containers and publish to Docker Hub. Testing Tools: Introduction to Selenium and its features, Java Script testing

UNIT-V:	**Configuration Management - ANSIBLE:**

Introduction to Ansible, Ansible tasks Roles, Jinja2 templating, Vaults, Deployments using Ansible. CONTAINERIZATION USING UBERNETES(OPENSHIFT): Introduction to Kubernetes Namespace & Resources, CI/CD - On OCP, BC, DC& Config Maps, Deploying Apps on Open shift
Container Pods. Introduction to Puppet master and Chef

Text Books

1. Joyner, Joseph., DevOps for Beginners: DevOps Software Development Method Guide for Software Developers and It Professionals, 1st Edition Mihails Konoplows, 2015.
2. Alisson Machado de Menezes., Hands-on DevOps with Linux,1st Edition, BPB Publications, India, 2021.

Reference Books

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10
2. Gene Kim Je Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona ,Joakim Practical DevOps,1stEdition,Packt Publishing,2016.
4. Joakim Verona. Practical Devops, Second Edition.In gram short title; 2nd edition (2018). ISBN10: 1788392574
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's View point. Wiley publications. ISBN:9788126579952

M Tech II Semester			L	T	P	C
			3	0	0	3
	SECURE CODING (Program Elective- III)					

Course Objectives: The main objectives of introducing this course are

1. Understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities.
2. Knowledge of outline of techniques for developing a secure application.
3. Recognize opportunities to apply secure coding principles.

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

CO1	Construct threat models by applying core security principles to a system's design.
CO2	Develop secure C code by implementing mitigation strategies for string and pointer vulnerabilities
CO3	Build secure C++ and Java programs that mitigate dynamic memory and integer security vulnerabilities.
CO4	Construct secure SQL queries and web forms to prevent injection and XSS attacks.
CO5	Develop security requirements using misuse cases and integrate security practices into the software lifecycle.

SYLLABUS	
UNIT-I:	**Introduction-**
Need for secure systems, Proactive security development process, Security principles to live by and threat modelling.	
UNIT-II:	**Secure Coding in C-**
Character strings- String manipulation errors, String Vulnerabilities and exploits Mitigation strategies for strings, Pointers, Mitigation strategies in pointer based vulnerabilities Buffer Overflow based vulnerabilities	
UNIT-III:	**Secure Coding in C++ and Java**
Dynamic memory management, Common errors in dynamic memory management, Memory managers, Double –free vulnerabilities, Integer security, Mitigation strategies	
UNIT – IV	**Database and Web Specific Input Issues-**
Quoting the Input, Use of stored procedures, Building SQL statements securely, XSS related attacks and remedies	

UNIT-V:	**Software Security Engineering-**

Requirements engineering for secure software: Misuse and abuse cases, SQUARE process model Software security practices and knowledge for architecture and design

Text Books

1.Writing Secure Code, 2 nd Edition, Michael Howard, David LeBlanc, Microsoft Press, 2003

Reference Books:

2. Secure Coding in C and C++, Robert C. Seacord, 2 nd edition, Pearson Education, 2013

3. Software Security Engineering: A guide for Project Managers, 1 st ed, Julia H. Allen, Sean J.Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, Addison-Wesley Professional,2008

M Tech II Semester			L	T	P	C
			0	1	2	2
	MACHINE LEARNING LAB					

Course Objectives: The main objectives of introducing this course are

1. To learn about computing central tendency measures and Data pre-processing techniques
2. To learn about classification and regression algorithms
3. To apply different clustering algorithms for a problem

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

CO1	Apply pre-processing techniques like attribute selection and discretization to prepare a raw dataset for analysis
CO2	Develop classification and regression models by utilizing supervised learning algorithms including KNN, Decision Trees, and Random Forests.
CO3	Build predictive models for both linear and logistic relationships by applying regression algorithms to solve real-world problems.
CO4	Utilize neural network and support vector algorithms to solve complex non-linear classification problems.
CO5	Solve unsupervised learning problems by building and evaluating clustering models using algorithms like K-Means and EM.

List of Experiments	
Experiment-1	Compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation.
Experiment-2	Apply the following Pre-processing techniques for a given dataset. a. Attribute selection b. Handling Missing Values c. Discretization d. Elimination of Outliers
Experiment-3	Apply KNN algorithm for classification and regression
Experiment-4	Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results
Experiment-5	Demonstrate decision tree algorithm for a regression problem
Experiment-6	Apply Random Forest algorithm for classification and regression

Experiment-7	Demonstrate Naïve Bayes Classification algorithm
Experiment-8	Apply Support Vector algorithm for classification
Experiment-9	Demonstrate simple linear regression algorithm for a regression problem
Experiment-10	Apply Logistic regression algorithm for a classification problem
Experiment-11	Demonstrate Multi-layer Perceptron algorithm for a classification Problem.
Experiment-12	Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.
Experiment-13	Demonstrate the use of Fuzzy C-Means Clustering
Experiment-14	Demonstrate the use of Expectation Maximization based clustering algorithm

M Tech II Semester			L	T	P	C
			0	1	2	2
	NATURAL LANGUAGE PROCESSING LAB					

Course Objectives: The main objectives of introducing this course are

1. Design Neural networks to solve real world problems
2. Build RNN, CNN models for classification
3. Choose appropriate pre-trained model to solve real time problem
4. Apply different NLP techniques using NLTK package
5. Design solutions to real world problems using NLP

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

CO1	Build and train deep learning models like MLPs and CNNs for classification.
CO2	Utilize RNNs and LSTMs to model sequential data for analysis.
CO3	Apply NLP techniques including embeddings and n-grams to process text.
CO4	Develop NER systems to solve information extraction tasks.
CO5	Build NLP pipelines to solve sentiment analysis problems.

List of Experiments	
Experiment-1	Implement Multilayer Perceptron algorithm for MNIST Handwritten Digit Classification.
Experiment-2	Design Neural Network for following problems i). Movie reviews classification (Binary Classification) using IMDB dataset. ii). News Wires classification (Multiclass Classification) using Reuters dataset.
Experiment-3	Implement a Recurrent Neural Network(RNN) and LSTM for IMDB movie review classification problem
Experiment-4	Build a Convolution Neural Network for simple image (dogs and Cats) classification
Experiment-5	Use a Pre-trained Convolution Neural Network LeNet, AlexNet for image classification
Experiment-6	Implement One Hot Encoding and Word Embeddings on any real world dataset
Experiment-7	Create Sample list at least 10 words POS tagging and find the POS for any given word
Experiment-8	Write a Python program to i). Perform Morphological Analysis using NLTK library

	ii)Generate n-grams using NLTK N-Grams library iii). Implement N-Grams Smoothing
Experiment-9	Write a program to implement Named Entity Recognition(NER) for any corpus
Experiment-10	Using NLTK package to convert audio file to text and text file to audio files
Experiment-11	Write a program to perform Auto-Correction of spellings for any text
Experiment-12	Implement twitter sentiment analysis using NLP.

Recommended MOOC Courses

S No	Name of the Course
1	Introduction to Industry 4.0 and Industrial Internet of Things
2	Ethical Hacking
3	Cyber Security and Privacy
4	Practical Cyber Security for Cyber Security Practitioners
5	Responsible & Safe AI Systems
6	Privacy and Security in Online Social Media
7	Reinforcement Learning
8	Computer Vision
9	Deep Learning for Computer Vision
10	Deep Learning-IIT RopAR

- Courses many be added based on the recommendations from the Department Advisory committee or the BOS from time to time.