National Institute of Technology Raipur

(An Institute of National Importance)

G.E. Road, Raipur – 492010 (C.G.)

Website: http://www. nitrr.ac.in **Phone:** +91-771-2254200

Department of Information Technology

MTech (IT) SEMESTER: I (New Scheme & Syllabus)

S.No.	Board of Studies	Sub. Code	Subject Name	Periods/week			Examination Scheme					Total Marks	Credits
				L	T	P	TA	FE	SE	ESE	Pract. ESE		
1	Information Technology	IT MC101	Distributed and Parallel System	4	0	-	20	15	15	100	-	150	4
2	Information Technology	IT MC102	Advanced Algorithms and Data Structures	4	0	-	20	15	15	100	-	150	4
3	Information Technology	IT MC103	Advanced Machine Learning	4	0	-	20	15	15	100	-	150	4
4	Information Technology	IT ME11X	Elective-1	4	0	-	20	15	15	100	-	150	4
5	Information Technology	IT ME12X	Elective-2	4	0	-	20	15	15	100	-	150	4
6	Information Technology	IT ML101	Lab 1 (Advanced Programming Lab)	-	-	3	75	-	-		50	125	2
7	Information Technology	IT ML102	Lab 2 (Advanced Machine Learning Lab)	-	-	3	75	-	-		50	125	2
			Total	**20**	**0**	**6**	**250**	**75**	**75**	**500**	**100**	**1000**	**24**

S.No	Elective-1	Elective-2
1	Mathematical Concepts of Computer Science	Natural language Processing
2	Biomedical Signal and Image Processing	Advanced Computer Architecture
3	Advanced Database Management System	Advanced Computer Network
4	Cloud Computing	Blockchain Technology
5	Quantum Computing and Algorithm	Bioinformatics
6	Advanced Software Design and Architecture	Computational geometry

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	Distributed and Parallel Computing		
Course – Code	ITME111		
Course – Periods / Week	(L + T + P) ↔(3 + 1 + 0)		
Course – Exam Scheme	(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)		
Course – Total Marks	150		
Course – Credits	4		
Course – Type	Elective – 1		

Course Outcomes:	
Students will be able to –	
CO-1	Understand the fundamentals and applications of Distributed Systems & other Computing paradigms.
CO-2	Know the synchronization mechanisms in Distributed Systems.
CO-3	Apply various Mutual Exclusion, Election and Distributed Consensus algorithms.
CO-4	Realize Distributed transaction and Distributed Storage mechanisms.
CO-5	Know Failure Recovery and Deadlock Detection algorithms in Distributed Systems.
CO-6	Explore and implement Parallel Computing and High-Performance Computing systems

Course Contents:	
UNIT-1	**Introduction to Distributed Computing:** Introduction to Distributed Systems and other Computing paradigms, advantages of Distributed Systems over Centralized systems, fundamental characteristics of Distributed Systems, types of Distributed Systems, design challenges and applications of Distributed Systems. Models of Distributed Systems: physical models, fundamental models and architectural models. Peer to Peer & Client-Server model, verities of Client-Server model.
UNIT-2	**Clocks in Distributed Computing:** Concept of clock in Distributed System, Clock synchronization. Physical clocks: GMT, TAI and UTC time. Computer clock, clock skew, clock drift. External and internal synchronization, Cristian's algorithm, Berkeley Algorithm, Network Time Protocol (NTP). Logical clocks: Causal Ordering, Lamport's Logical Clock, Vector Clocks, global state, consistent and inconsistent cut, Chandy and Lamport's algorithm to construct consistent cut.

UNIT-3	**Mutual Exclusion, Election, and Consensus in Distributed Systems:** Distributed Mutual Exclusion, Critical Section and requirements for Mutual Exclusion, Mutual Exclusion Algorithms: Central Server algorithm, Ring-based algorithm, Ricart and Agrawala's algorithm. Election Algorithms: Election algorithm requirements, Chang and Robert's Ring-based algorithm, Bully algorithm. Distributed Consensus: Flood-set algorithm, Byzantine failure, Byzantine General Problem.
UNIT-4	**Distributed Transactions & Recovery:** Distributed Transaction, Distributed Storage, Replication, Recovery in Distributed System. Commit protocols: 1-Phase, 2-Phase and 3-Phase commit protocols, Distributed Deadlock Detection.
UNIT-5	**Parallel Computing:** Introduction to parallel computing, Flynn's classification. Parallel Computing topologies: Chain, Ring, Mesh, Torus, tree, Star, Cube, Hyper tree, etc. Uniform Memory Access (UMA) & Non uniform Memory Access (NUMA), Multi-processor System, Parallel Matrix Multiplication. Introduction to High-Performance Computing (HPC), architecture of a supercomputing system, Parallel algorithms paradigm. Introduction to OpenMP & MPI programming model, applications of HPC systems.

Reference Books:

1.	M. Singhal, N. G. Shivaratri, N. Shivaratri, ***Advanced Concepts in Operating Systems***, McGraw-Hill Science, 1st edition, 1994
2.	P. K. Sinha, ***Distributed Operating Systems***, PHI, 2012.
3.	Michel J. Quinn, ***Parallel Computing: Theory and Practice***, McGraw-Hill Science, 2nd edition, 2017.
4.	G. Couloris, J. Dollimore, T. Kindberg, ***Distributed Systems: Concepts & Design***, Pearson Education India, 2010.
5.	Tanenbaum Andrew S., **Distributed Systems**, 3rd edition, CreateSpace, 2017.
6.	A. D. Kshemkalyani, M. Singhal, ***Distributed Computing: Principles, Algorithms, and Systems***, Cambridge, 2008
7.	G. Hager, G. Wellein, ***Introduction to High Performance Computing for Scientists and Engineers***, CRC Press; 1st edition, 2010

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	**Advanced Algorithms and Data Structures**		
Course – Code	**ITMC102**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Core**		

Course Outcomes:

Students will be able to –

CO-1	Analyze worst-case running times of algorithms using asymptotic analysis.
CO-2	Classify problems into distinct complexity classes based on deterministic algorithms, approximation algorithms, and parameterized algorithms.
CO-3	Analyze the complexity of graph problems and apply it to different real word problems.
CO-4	Examine approximation algorithms and ascertain their approximation factor.
CO-5	Design and analyze efficient randomized algorithms.

Course Contents:

UNIT-1	**Foundations of Algorithms:** Algorithm Analysis, Asymptotic Notations (Big-O, Big-Theta, Big-Omega), Worst-case, Average-case, and Best-case Analysis, Amortized Analysis. Algorithm design techniques: Divide and Conquer Algorithms: Merge Sort, Quick Sort, Master Theorem and its applications; Dynamic Programming: Principles of Dynamic Programming, Examples: Fibonacci Series, Knapsack Problem; Greedy Algorithms: Principles and applications, Minimum Spanning Tree (Prim's and Kruskal's algorithms), Dijkstra's algorithm.
UNIT-2	**Advanced Data Structures:** Advanced Trees: AVL Trees, Red-Black Trees, B-Trees and B+ Trees, Heaps and Priority Queues, Binary Heaps, Heap Operations, Applications in algorithms (e.g., Dijkstra's algorithm), Hashing: Hash Functions and Hash Tables, Collision Resolution Techniques (Chaining, Open Addressing), Graph Algorithms: Graph Representation (Adjacency Matrix, Adjacency List, Graph Traversal (DFS, BFS), Shortest Path Algorithms (Dijkstra's, Bellman-Ford), Minimum Spanning Tree (Prim's, Kruskal's).
UNIT-3	**Advanced Algorithm Techniques:** Advanced Divide and Conquer, Strassen's Matrix Multiplication, Closest Pair of Points.

	Randomized Algorithms: Randomized Quick Sort, Las Vegas and Monte Carlo algorithms, Approximation Algorithms: Principles and examples (e.g., Set Cover, Traveling Salesman Problem), Parallel and Distributed Algorithms: Parallel Sorting Algorithms, Distributed Graph Algorithms.
UNIT-4	**Advanced Topics in Data Structures:** Tries Structures and Suffix Trees: Applications in text processing, Advanced Hashing Techniques: Perfect Hashing, Bloom Filters Advanced Tree Structures: Segment Trees, Fenwick Trees (Binary Indexed Trees), Persistent Data Structures: Basics and applications.
UNIT-5	**String Algorithms:** Pattern Matching (KMP algorithm, Rabin-Karp algorithm), Longest Common Subsequence, Geometric Algorithms: Convex Hull Algorithms (Graham Scan, Jarvis March), Network Flow Algorithms: Maximum Flow (Ford-Fulkerson algorithm, Edmonds-Karp algorithm), Minimum Cut, NP-Hard and NP-Complete Problems, Approximation Schemes.

Reference Books:

1.	D. C. Kozen, ***The Design and Analysis of Algorithms***, Springer, 1992.
2.	T. H. Cormen, C. E. Leiserson, and R. L. Rivest, ***Introduction to Algorithms***, PHI, 1990.
3.	R. Motwani and P. Raghavan, ***Randomized Algorithms***, Cambridge University Press, 1995.
4.	C. H. Papadimitriou, ***Computational Complexity***, Addison Wesley, 1994.
5.	***Algorithmic Graph Theory and Perfect Graphs***, Martin Charles Golumbic, 2004, Elsevier, Second Edition.
6.	***Treewidth: Computations and Approximations***, Ton Kloks, Springer-Verlag, 1994.
7.	***Algorithms and Complexity***, Herbert S. Wilf, AK Peters/CRC Press, 2002, Second Edition
8.	***Parameterized Complexity***, Rodney G. Downey and M. R. Fellows, Springer, 2012.
9.	***Approximation Algorithms***, Vijay V. Vajirani, Springer, 2001

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	**Advanced Machine Learning**		
Course – Code	**ITMC103**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Core**		

Prerequisites:

- Data Structures
- Fundamental of Computer concepts

Course Outcomes:

Students will be able to –

CO-1	To understand the fundamental principles that allow one to design machine learning algorithms.
CO-2	To become familiar with specific, widely used machine learning algorithms.
CO-3	To introduce building blocks of deep neural network architecture.
CO-4	To understand representation and transfer of knowledge using deep learning.
CO-5	To learn to use deep learning tools and framework for solving real-life problems.

Course Contents:

UNIT-1	**Introduction to Machine Learning Framework:** Machine Learning Pipeline, Key Terminology and tasks, Types of Machine Learning Techniques, Performance Evaluation, Bias-Variance Trade-off, Python machine learning environments **Data Pre-processing:** Types of features, Datatransformation, discretization, Data manipulation, standardization and Data Normalization
UNIT-2	**Feature Selection Techniques:** Filter-based techniques, Wrapper-based techniques, Embedded-based techniques,**Dimension reduction:** Principal component analysis (PCA), LDA, **Predictive Analysis**: Linear regression, Multiple Linear regression.
UNIT-3	**Supervised, Unsupervised and Semi-supervisedLearning** **Supervised Learning:** K-NN, Naive Bayes, SVM,Logistic Regression, Decision Trees,

	Randon Forest, **Ensemble methods:** Bagging, Boosting, Stacking, Voting-based techniques. **Unsupervised Learning:** K-Mean Clustering Model, Semi-supervised Learning.
UNIT-4	**Fundamental concept of Artificial Neural Network (ANN) and Deep learning (DL):** Introduction to the Neural Network**,** Training Feed-Forward Neural Networks **Convolutional Neural Network (CNN)Theory:** Convolution Operation, Pooling Operation, Base CNN Architectures, Loss Functions, Activation Functions, Optimization Algorithms (e.g., Stochastic Gradient Descent, Adam), Regularization Techniques (e.g., Dropout, L1/L2 Regularization), Batch Normalization and Dropout.
UNIT-5	**Advanced DL Architecture:** RNN, LSTM, GRU, BiGRU, Autoencoder, Generative Adversarial Networks (GAN), Graph CNN, Federated Learning.

Reference Books:

1.	Trevor Hastie, Robert Tibshirani, Jerome Friedman "**The Elements of Statistical Learning**: Data Mining, Inference, and Prediction" (2nd Edn.), Springer, 2014.
2.	Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, "***Introduction to Data Mining*",**2nd Edition, 2019, Pearson.
3.	James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. "***An introduction to statistical learning****: With applications in python***",** Springer Nature, 2023.
4.	*Ian Goodfellow, Yoshua Bengio, Aaron Courville, "**Deep Learning**", The MIT Press, 2016*
5.	*Aurélien Géron, "**Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**", O'Reilly Media.*
6.	*Francois Chollet, "**Deep Learning with Python**", Manning Publications, 2017*
7.	Buduma, N., and Nicholas Locascio. "***'Fundamentals of Deep Learning: Designing Nextgeneration Machine Intelligence Algorithms***' 2017 O'Reilly Media"
8.	Soledad Galli "***Python Feature Engineering Cookbook***",Packt Publication,2020.

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	Mathematical Concepts of computer Science		
Course – Code	ITMC101		
Course – Periods / Week	(L + T + P) ↔(3 + 1 + 0)		
Course – Exam Scheme	(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)		
Course – Total Marks	150		
Course – Credits	4		
Course – Type	Core		

Course Outcomes:

Students will be able to –

CO-1	Apply the concepts of Linear Algebra.
CO-2	Use the Mathematical concepts to solve real world problems.
CO-3	Apply the concepts of Probability and Distribution for a given problem.
CO-4	Use the concepts of Discrete Mathematics for solving the given problem.
CO-5	Apply the concept of reduction to prove given problem is hard.

Course Contents:

UNIT-1	**Discrete Mathematics:** Sets and Relations, Mathematical Logic and Induction, Elementary Combinatorics, Recurrence Relations, Lattices as Partially Ordered Sets, Graphs, Trees. Groups, Rings and Fields.
UNIT-2	**Linear Algebra:** System of linear equations, Matrices, Solving Systems of Linear Equations, Vector Spaces, Linear Independence, Basis and Rank, Vector spaces and subspaces, Linear transformations, Eigenvalues and eigenvectors, Orthogonality and orthogonal projections, Singular value decomposition, Applications to computer graphics, machine learning, and cryptography .
UNIT-3	**Calculus:** Limits and continuity, Differentiation and integration, Multivariable calculus (partial derivatives, gradients, multiple integrals), Optimization techniques (unconstrained and constrained).
UNIT-4	**Probability and Statistics:** Probability theory (basic concepts, random variables, distributions, Statistical inference (estimation, hypothesis testing), Regression analysis, Bayesian methods, Bayes Theorem, Summary Statistics and Independence, Markov Chain.

UNIT-5	**Complexity Theory:** Basics of algorithm analysis (time and space complexity), P vs NP problem, NP-completeness and reductions, Approximation algorithms.
Reference Books:	
1.	. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong***, Mathematics for Machine Learning***, Cambridge University Press, 2020. (for topics other than Discrete Mathematics)
2.	Joe L. Mott, Abraham Kandel, Theodore P. Baker, ***Discrete Mathematics for Computer Scientists and Mathematicians***, Second Edition, PHI, 2001.
3.	J. P. Tremblay and R. Manohar, ***Discrete Mathematical Structures with Applications to Computer Science***, MGH, 1997.
4.	Kenneth H. Rosen, ***Discrete Mathematics and its Applications with Combinatorics and Graph Theory***, Seventh Edition, MGH, 2011
5.	"***Introduction to Graph Theory***" by Douglas B. West
6.	***"Introduction to Linear Algebra"***by Gilbert Strang

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	**Biomedical Signal and Image Processing**		
Course – Code	**ITME112**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective – 1**		

Prerequisites:

- Basic knowledge of signal processing concepts.
- Familiarity with linear algebra and calculus.
- Introduction to programming (preferably in MATLAB or Python)
- Fundamental understanding of biomedical engineering principles

Course Outcomes:

Students will be able to –

CO-1	Demonstrate an understanding of the fundamentals of biomedical signal and image processing.
CO-2	Apply various signal processing techniques to analyze biomedical signals.
CO-3	Explain the principles and methods of medical image formation and enhancement.
CO-4	Implement advanced algorithms for the analysis and interpretation of medical images.
CO-5	Utilize software tools to perform practical biomedical signal and image processing tasks.

Course Contents:

UNIT-1	**Introduction to Biomedical Signal Processing:** Overview of Biomedical Signals, Signal Acquisition and Sampling, Noise and Artifacts in Biomedical Signals, Time-Domain Analysis of Biomedical Signals, Frequency-Domain Analysis of Biomedical Signals, Case Studies: ECG, EEG, and EMG Signal Processing.
UNIT-2	**Advanced Signal Processing Techniques**: Digital Filtering and Signal Enhancement, Wavelet Transform and Multiresolution Analysis, Adaptive Filtering, Signal Compression Techniques, Feature Extraction and Pattern Recognition, Case Studies: Signal Processing in Biomedical Instrumentation.

UNIT-3	**Introduction to Biomedical Image Processing:** Basics of Image Formation, Image Acquisition Techniques, Image Representation and Formats, Image Enhancement Techniques, Noise Reduction in Medical Images, Case Studies: Enhancing X-ray, MRI, and CT Images. .
UNIT-4	**Advanced Image Processing and Analysis:** Image Segmentation Techniques, Edge Detection and Feature Extraction, Image Registration and Fusion, Morphological Image Processing, Texture Analysis and Classification, Case Studies: Image Analysis in Medical Diagnosis.
UNIT-5	**Applications and Tools in Biomedical Signal and Image Processing:** Applications of Signal Processing in Healthcare, Advanced Algorithms in Medical Imaging, Machine Learning and Deep Learning in Biomedical Imaging, Software Tools for Signal and Image Processing (MATLAB, Python), Practical Implementation and Case Studies, Case Studies: Recent Advances in Biomedical Imaging Technology. .

Reference Books:

1.	***Biomedical Signal Analysis*** : Rangarajan M Rangarajan.
2.	Gonzalez, R., and R. E. Woods. ***Digital Image Processing***. 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002. ISBN: 9780201180756 .
3.	Rabiner, L. R., and R. W. Schafer. ***Digital Processing of Speech Signals***. Upper Saddle River, NJ: Prentice-Hall, 1978. ISBN: 9780132136037.
4.	***Essentials of medical physiology***: K. Sembulingam.
5.	***DSP Principles and Algorithm***: Proakis.
6.	***Principles of Neurology***: Allan H. Ropper
7.	***EEG ERP Analysis Methods and Application***:: Nidal Kamel

Name of Program	M. Tech.	Semester – I	Year – I
Course– Name	**Advanced Database Management System**		
Course– Code	**ITME113**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1+ 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective – 1**		

Prerequisites:

- Database System Concepts

Course Outcomes:

Students will be able to –

CO-1	Understand the basic concepts and terminology related to distributed DBMS and its design.
CO-2	Design and develop query processing strategies.
CO-3	Understand transaction processing and concurrency control in distributed databases.
CO-4	Understand reliability and replication concepts in distributed databases

Course Contents:

UNIT-1	**Overview of Distributed Database and Distributed Database Design** Distributed Database Management Systems - Promises of distributed database, design issues of distributed databases, distributed database architecture, Distributed Database Access Primitives, Integrity Constraints in Distributed Databases, Data fragmentation, horizontal fragmentation, vertical fragmentation, Allocation of Fragments, allocation problem, allocation model, Translation of Global Queries to Fragment Queries, The Equivalence Transformation for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping - Aggregate Function Evaluation, Parametric Queries, Database Integration, Schema Matching, Schema Integration, Schema Mapping.
UNIT-2	**Query Decomposition and Data Localization** Overview of Query Processing-objectives, Characterization of Query Processors, Layers of Query Processing, Query Decomposition and Data Localization- Localization of Distributed Data, Optimization of Distributed Queries, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization.

UNIT-3	**Distributed Transaction Management and Concurrency Control** Introduction to Transaction Management, Properties of Transactions, Types of Transactions, Distributed Concurrency Control, Taxonomy of Concurrency Control Mechanisms, Locking Based Concurrency Control Algorithms, Timestamp Based Concurrency Control Algorithms, Optimistic Concurrency Control Algorithms, Deadlock Management - The System R* ,The Architecture of System R*, Compilation - Execution and Recompilation of Queries, Protocols for Data Definition and Authorization in R*, Transaction and Terminal Management.
UNIT-4	**Reliability** Distributed DBMS Reliability, Reliability Concepts and Measures, Failures in Distributed DBMS, Local Reliability Protocols, Distributed Reliability Protocols.
UNIT-5	**Replication and Current Trends** Data Replication, Consistency of Replicated Databases, Update Management Strategies, Replication Protocols, Current trends in No SQL, New SQL data management issues on the cloud, Stream data management.

Reference Books:	
1.	Stefano Ceri, Guiseppe Pelagatti, “Distributed Databases - Principles and Systems”, Tata McGraw Hill, 2008.
2.	Ozsu M.T., Sridhar S., “Principles of Distributed database systems”, Pearson education, 2011.
3.	Korth & Sudarshan, “Database system concept”, Tata McGraw Hill, 2008. Optional Materials:
4.	Raghu Rama Krishnan, Johnaas Gehrke, “Database Management Systems”, TataMcGrawHill, 2000.
5.	Elmasri, Navathe, “Fundamentals of Database Systems”, Addison-Wesley, Fifth Edition 2008.

Name of Program	M. Tech.	Semester – I	Year – I

Course – Name	**Cloud Computing**
Course – Code	**ITME114**
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**
Course – Total Marks	**150**
Course – Credits	**4**
Course – Type	**Elective – 1**

Course Outcomes:

Students will be able to –

CO-1	Understand the fundamentals and applications of Computing paradigms.
CO-2	Know the Cloud Deployment and Service Delivery Models.
CO-3	Understand Virtualization and Containers.
CO-4	Apply various VM Migration techniques.
CO-5	Build, Simulate and deploy a Cloud Service.
CO-6	Identify and select the right Cloud Migration Strategy.
CO-7	Analyse the Importance of Cloud and Data Security.

Course Contents:

UNIT-1	**Introduction to Cloud Computing:**Introduction to Cloud Computing, NIST view of Cloud Computing, Various Computing Paradigms, Cloud Computing Architecture, Cloud Computing Characteristics, Benefits and Challenges of Cloud Computing, Opportunities and Challenges of Cloud Computing, Advantages and Disadvantages of Cloud Computing.
UNIT-2	**Cloud Deployment and Service Delivery Models:**Introduction to Cloud Deployment Models, Types of Cloud Deployment Models, Examples of Private, Public, Hybrid and Community Cloud. Comparison of Cloud Deployment Models. Overview of Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) model. SaaS Maturity Model, Serverless Computing. Use cases of SaaS, PaaS, and IaaS. Advantages and Disadvantages of SaaS, PaaS and IaaS. Related products and Services of SaaS, PaaS and IaaS. .
UNIT-3	**Virtualization and VM Migration:**Introduction to Virtualization, Comparison of Traditional IT Infrastructure with Virtualized Infrastructure, Virtual Machines, Types

	of Virtualizations, Hypervisor, Types of Hypervisors, Architecture of Xen Hypervisor, Architecture of KVM Hypervisor, Benefits of Virtualization, Cloud Computing vs Virtualization. Containers, Microservices, Benefits of Containers, VMs vs Containers vs Serverless Computing. Virtual Machine Migration, Virtual Machine Migration Services, Pre-copy Migration, Post-copy Migration. Load balancing in Cloud Computing.
UNIT-4	**Cloud Computing Platforms and Tools:** Overview of OpenStack, Amazon Web Services, Microsoft Azure, Google Cloud, CloudSim and Aneka. Installing a Cloud service using Eucalyptus, Open Nebula, OpenStack, Amazon Web Services, Microsoft Azure, Google App Engine. Building of a private cloud using open-source tools. Understanding various cloud plugins, Setting up a cloud environment. Working of CloudSim tool and Aneka Cloud Computing Platform.
UNIT-5	**Cloud Migration, Cloud Security and Cloud enabling Technologies:**Cloud Migration, Benefits of Cloud Migration, Cloud Migration Challenges, Cloud Migration Strategies: The 6 R's of cloud migration. Cloud Security, Importance of Cloud Security, Infrastructure Security, Network level security, Host level security, Application-level security, Data security and Storage, Data privacy and security Issues, Jurisdictional issues raised by Data location, Identity & Access Management, Access Control, Authentication in cloud computing, Client access in cloud, Cloud contracting Model.

Reference Books:

1.	George Reese, ***Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice)***, O'Reilly, 2009.
2.	Rajkumar Buyya, James Broberg, Andrzej Goscinski, ***Cloud Computing: Principles and Paradigms***", Wiley.
3.	Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, ***Cloud Computing: A Practical Approach***, McGraw Hill Education, 2017.
4.	Rajkumar Buyya, S. Thamarai Selvi, Christian Vecchiola ***Mastering Cloud Computing: Foundations and Applications Programming***, Morgan Kaufmann, 2013
5.	Barrie Sosinsky ***Cloud Computing Bible***, Wiley India Pvt Ltd, 2011.
6.	Thomas Erl, Zaigham Mahmood, Ricardo Puttini, ***Cloud Computing: Concepts, Technology & Architecture***, Pearson Education, 2013

Name of Program	**M. Tech.**	**Semester – I**	**Year – I**

Course – Name	**Quantum Computing and Algorithms**
Course – Code	**ITME115**
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**
Course – Total Marks	**150**
Course – Credits	**4**
Course – Type	**Elective – 1**

Prerequisites:

- Linear Algebra & Probability Theory
- Discrete Mathematical Structures
- Basics of Quantum Mechanics
- Programming Basics and Skills

Course Outcomes:

Students will be able to –

CO-1	Realize the potential of quantum parallelism and computing principles for solving computationally intensive problems.
CO-2	Develop quantum algorithms using several design strategies to efficiently solve the existing problems.
CO-3	Analyze performance of quantum algorithms outperforming classical algorithms.
CO-4	Simulate and implement the quantum circuits equivalent to quantum algorithms.
CO-5	Investigate the quantum algorithms as effective solutions to solve the real-world problems in diversified domains of computational science and technology.

Course Contents:

UNIT-1	**Quantum Computing Basics:** Introduction and Overview, Quantum Mechanics Fundamentals, Quantum Postulates, Quantum Parallelism and Correlations, Concept of Qubit, Quantum Information Processing, Quantum Computer Science.
UNIT-2	**Quantum Circuits Design:** Classical and Quantum Computation Model, Quantum State Transformation, Basic Quantum Gates, Quantum Circuits. Quantum-Based Circuit Designs of Classical Computation, Applications of Quantum Gates.
UNIT-3	**Quantum Design Strategies:** Quantum Parallelism (Revisited), Quantum Oracle Approach, Quantum Fourier Transform (QFT), Quantum Phase Estimation (QPE), Quantum Amplitude Amplification (QAA), Quantum Walk, Adiabatic Quantum

	Computing, Hybrid Classical-Quantum Algorithms, Quantum Error Correction.
UNIT-4	**Quantum Algorithms:** Introduction to Quantum Algorithms, Speedup Analysis, A Few Quantum Algorithms: Deutsch, Deutsch-Jozsa, Bernstein Vazirani, Simon's, Grover's, Shor's algorithms, and Quantum Computational Complexity Theory.
UNIT-5	**Quantum Programming and Realization:** Understanding Physical Realization of Quantum Computer Technology, Quantum Programming using QuEST, Quirk and IBM Qiskit and etc. Quantum Algorithmic Simulation and Emulation. Significant Research and Development of Quantum Computing in Computer Science.

Reference Books:

1.	Michael A. Nielsen & Isaac L. Chuang, ***Quantum Computation and Quantum Information***, 10th Anniversary Edition – 2010, Cambridge University Press.
2.	Phillip Kaye, ***An Introduction to Quantum Computing,*** First Published Edition – 2007, Oxford University Press.
3.	Noson S. Yanofsky and Micro A. Mannucci. ***Quantum Computing for Computer Scientists***, First Published Edition – 2008, Cambridge University Press.
4.	N. David Mermin, ***Quantum Computer Science – An Introduction***, I Edition – 2007, Cambridge University Press.
5.	Ivan B. Djordjevic, Quantum Information Processing, ***Quantum Computing, and Quantum Error Correction – An Engineering Approach***, Second Edition – 2021, Academic Press – An Imprint of Elsevier.

Name of Program	**M. Tech.**	**Semester – I**	**Year – I**

Course – Name	**Advanced Software Design and Architecture**
Course – Code	**ITME116**
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**
Course – Total Marks	**150**
Course – Credits	**4**
Course – Type	**Elective – 1**

Course Outcomes:

Students will be able to –

CO-1	To understand the objectives of Software Architectural design.
CO-2	To learn various Software Architecture prevalent wrtMVC, SOA etc.

Course Contents:

UNIT-1	**Introduction to Algorithm Design, Software Design and Architecture:** What are Software system Design objectives, purpose and approaches of efficient system design, and Functional Independence in Software Design with Coupling and Cohesion? Overview of OO Design: Class Diagrams Object Diagrams. Sequence and Collaboration Diagrams, Static and dynamic modelling approaches for efficient design
UNIT-2	**UNIT -2: Software Design Principles:** Role of Modelling and Design, Design Metrics, OO Software Design. Design Principles with Applications. Iterative Refinement Behaviour, Iterative Refinement Minimalism. Mobile Software and Design: Characteristics and Requirements, Mobile Interaction designs, UX design.
UNIT-3	**Design Patterns and Architectural Consideration:**Recent Trends in Software Design with a special focus on Mobile Apps Development. The GoF and Evolution of Design Patterns.
UNIT-4	**Pattern-based Design:**Design Patterns with Creational Design Patterns, Structural Design Patterns and Behavioural Design Patterns. Example elaborations of popular patterns with suggested areas of Applications.Recent Trends in Software Architecture: Cloud-Based Architecture, Service-Oriented Architecture etc.

Reference Books:

1.	G. Booch, ***"Object-Oriented Analysis and Design with Applications"*** 2nd Edition, PHI, New Delhi
2.	Ralph Johnson, John Vlissides, Richard Helm, and Erich Gamma,***"Design Patterns"***.

3.	Richard N. Taylor et al., ***"Software Architecture: Foundations, Theory, and Practice"*** John Wiley and Sons.

Name of Program	M. Tech.	Semester – I	Year – I
Course– Name	**Natural Language Processing**		
Course– Code	**ITME121**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective – 2**		

Prerequisites:

- Data Structures
- Machine Learning
- Basics of Probability

Course Outcomes:

Students will be able to –

CO-1	justify the various steps necessary for processing natural language
CO-2	suggest appropriate lexical and parsing techniques for a given natural language
CO-3	apply appropriate statistical models for a given natural language application
CO-4	modify existing algorithms to suit any natural language for processing
CO-5	suggest appropriate pre-processing steps essential for the various applications involving natural language processing

Course Contents:

UNIT-1	**Lexical Analysis Lexical Analysis** Regular expression and Automata for string matching - Words and Word Forms - Morphology fundamentals - Morphological Diversity of Indian Languages - Morphology Paradigms - Finite State Machine / Transducers Based Morphology - Automatic Morphology Learning - Parts of Speech - N-gram Models - Hidden Markov Models.
UNIT-2	**Speech Processing Biology of Speech Processing** Place and Manner of Articulation - Word Boundary Detection - Argmax based computations - HMM and Speech Recognition - Text to Speech Synthesis - Rule based- Concatenative based approach.
UNIT-3	**Parsing Theories** Parsing Algorithms – Earley Parser - CYK Parser - Probabilistic Parsing - CYK -

	Resolving attachment and structural ambiguity - Shallow Parsing - Dependency Parsing - Named Entity Recognition - Maximum Entropy Models - Conditional Random Fields.
UNIT-4	**Lexical Knowledge Networks Meaning** Lexical Knowledge Networks - Wordnet Theory - Indian Language Wordnets and Multilingual Dictionaries - Semantic Roles - Word Sense Disambiguation - WSD and Multilinguality - Metaphors - Coreference and Anaphora Resolution.
UNIT-5	**Applications** Sentiment Analysis - Text Entailment - Machine Translation - Question Answering System - Information Retrieval - Information Extraction - Cross Lingual Information Retrieval (CLIR).

Reference Books:

1.	"**Speech and Language Processing**" by Daniel Jurafsky and James H. Martin (3rd edition)
2.	Foundations of Statistical Natural Language Processing, Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.
3.	"The Oxford Handbook of Computational Linguistics" edited by Ruslan Mitkov
4.	"Introduction to Information Retrieval" by Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	**Advanced Computer Architecture**		
Course – Code	**IT ME122**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective – 2**		

Prerequisites:

- Basic knowledge of computer organization and architecture
- Understanding of digital logic design
- Familiarity with programming (preferably in C or assembly language)
- Knowledge of fundamental concepts in operating systems and computer networks

Course Outcomes:

Students will be able to –

CO-1	Understand and articulate advanced computer architecture concepts.
CO-2	Evaluate and optimize various levels of the memory hierarchy.
CO-3	Implement and enhance instruction-level parallelism in processors.
CO-4	Assess and improve thread-level parallelism in shared-memory multicore systems.
CO-5	Develop and optimize data-level parallelism using modern GPU programming techniques

Course Contents:

UNIT-1	**Fundamentals of Advanced Computer Architecture:** Evolution of Computer architecture, system attributes to performance, Multi processors and multi computers, Multi-vector and SIMD computers, PRAM and VLSI models-Parallelism in Programming, conditions for Parallelism-Program Partitioning and Scheduling-program flow Mechanisms-Speed up performance laws-Amdahl's law, Gustafson's law-Memory bounded speedup Model.
UNIT-2	**Advanced Memory Hierarchy Design:**Memory Hierarchies, Cache Performance Optimization Techniques, Advanced Memory Technologies, Virtual Memory Management, Memory Hierarchy Design Considerations, Tools: Pin Instrumentation and Cachegrind, Case Study: Memory Hierarchies in Modern Processors.

UNIT-3	**Enhancing Instruction Level Parallelism (ILP):**Introduction of ILP, Concepts and Challenges of ILP, Compiler Techniques for ILP, Branch Prediction and Speculation Techniques, Dynamic Scheduling and Out-of-Order Execution, Techniques for Instruction Delivery, Limitations and Challenges of ILP, Tools: Modeling ILP with Pin Tool, Case Study: ILP in Intel Core i7 and ARM Cortex-A8.
UNIT-4	**Advanced Thread Level Parallelism (TLP):**Introduction to TLP, Architectures of Shared-Memory Multicore Systems, Performance Metrics for Multicore Systems, Cache Coherence and Synchronization Protocols, Memory Consistency Models, Multithreaded Programming Techniques using OpenMP, Case Study: TLP in Intel Skylake and IBM Power8.
UNIT-5	**Data Level Parallelism (DLP) and GPUs:**Introduction to DLP, Vector Processing and SIMD Architectures, GPU Architectures and Programming Models, GPU Memory Hierarchy and Optimization Techniques, Detecting and Exploiting Loop-Level Parallelism, CUDA and OpenCL Programming, Case Study: DLP in Nvidia Maxwell and Modern GPUs.

Reference Books:

1.	J.L. Hennessy and D.A. Patterson. ***Computer Architecture: A Quantitative Approach. 5th Edition***, Morgan Kauffmann Publishers, 2012.
2.	Hwang, K.***"Advanced computer architecture with parallel programming"***, McGraw Hill.
3.	J.P. Shen and M.H. Lipasti. ***Modern Processor Design: Fundamentals of Superscalar Processors.*** McGraw-Hill Publishers, 2005.
4.	D.B. Kirk and W.W. Hwu. ***Programming Massively Parallel Processors. 2nd Edition***, Morgan Kauffmann Publishers, 2012.

Name of Program	M. Tech.	Semester – I	Year – I

Course – Name	**Advanced Computer Network**
Course – Code	**ITME123**
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**
Course – Total Marks	**150**
Course – Credits	**4**
Course – Type	**Elective – 2**

Prerequisites:

- Fundamental of Computer Architecture and Operating System
- Networking

Course Outcomes:

Students will be able to –

CO-1	Understand advanced concepts and next generation networks
CO-2	Analyse TCP/IP variants, network Algorithm's, Protocols and their functionalities
CO-3	Comprehend features of SDN and its application to next generation systems
CO-4	Analyse the performance of various server implementations
CO-5	Understand advanced concepts and next generation networks

Course Contents:

UNIT-1	Packet-Switched Networks, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol Layers and Their Service Models, High Performance Switching and Routing: Introduction, performance considerations, IP address lookup, Algorithms for IP address lookup and optimization, hardware implementation of address lookup.
UNIT-2	**Network Layer:** Network Layer Issues- Routing, Congestion control-Internetworking - Issues, Address Learning Bridges, Spanning tree, Source routing, Bridges, Routers, Gateway.
UNIT-3	Introduction to the Link Layer, Error-Detection and -Correction Techniques, Switched Local Area Networks, Link Virtualization
UNIT-4	Software Defined Network -Comparison between SDN and traditional networks -SDN controller, Switch design, SDN Controller-Switch Protocols, Open Flow Protocol, Control Overhead & Handoff algorithms. Network Function Virtualization -NFV

	Architecture, Use cases, NFV Orchestration and NFV for 5G.
UNIT-5	Cloud Networking, Characteristics of Cloud Networking, Cloud Data Center Networking Topologies, Data Center Networking, Data Center Architectures
Reference Books:	
1.	Larry Peterson and Bruce Davie,***"Computer Networks: A Systems Approach"***,2022.
2.	James F. Kurose and Keith W. Ross, ***"Computer Networking: A Top-Down Approach Featuring the Internet"***, 2023.
3.	Eiji Oki, Roberto Rojas-Cessa, Christian Vogt, ***"Advanced Internet Protocols, Services and Applications"***, John Wiley and Sons Ltd, 2012.
4.	***High Performance Switches and Routers***, H. Jonathan Chao, Bin Liu, 2007, John Wiley & Sons, Inc. ISBN-10: 0-470-05367-4
5.	***Cloud Networking: Understanding Cloud-based Data Centre Networks,*** Gary Lee (Author), Morgan Kaufmann (Publisher), 2014, ISBN-139780128007280

Name of Program	M. Tech.	Semester – I	Year – I
Course – Name	**Blockchain Technology**		
Course – Code	**ITME124**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective-2**		

Prerequisites:

- Basic knowledge of programming
- Data Structures
- Networking
- Web Technology

Course Contents:

UNIT-1	**Fundamental concept of Blockchain Technology:** Origin of blockchain technology, Block Structure, Blockchain Layer Architecture, Blockchain Functional Architecture, Generic Elements of Blockchain, Blockchain infrastructure, Basic concepts of Peer-2-Peer Network, Types of Nodes, Types of Blockchain, Characteristics of Blockchain, Evolution of Blockchain Technology, Bitcoin Cryptocurrency.
UNIT-2	**Cryptography Primer for Blockchain:** Symmetric Cryptography Algorithms, Asymmetric Cryptography Algorithms: RSA, Elliptic Curve Cryptography, Homomorphic Cryptography, Cryptographic Hash Functions, Properties of cryptographic hash functions, SHA-256 hash function, Merkle Trees: Construction of Merkle Trees, Benefits of Merkle Trees, Digital Signatures, Zero-Knowledge Proof.
UNIT-3	**Consensus Algorithms for Blockchain:** Taxonomy of Consensus Algorithms, Proof-Based Consensus Algorithms, Vote-Based Consensus Algorithms, PAXOS consensus algorithm, Byzantine Fault Tolerance-Based Consensus, Practical Byzantine Fault Tolerance (PBFT), Delegated Byzantine Fault Tolerance (DBFT), Federated Byzantine Fault Tolerance (FBFT), Proof of Work (PoW), Proof of Stake (PoS), Hybrid Consensus: Proof-of-Burn (POB), Proof-of-Importance (POI).
UNIT-4	**Fundamentals of Ethereum Blockchain, Smart Contract,** Fundamentals of Solidity Programming**,** Building Blocks of Solidity, Understanding Data Types in Solidity, Control Flow Statements, Functions, Visibility and Modifiers, Inheritance and Libraries, Building a Simple Smart Contract Example.

UNIT-5	**Enterprise Blockchain:** Hyperledger Fabric Platform, Architecture of Hyperledger Fabric, Fundamentals of Tokenization Concept, Non-Fungible Tokens (NFTs), Use cases of Blockchain: E-Governance, Traceability and Anti-Counterfeiting
Reference Books:	
1.	Ambadas Tulajadas Choudhari, A. Sarfarz Ariff, Sham M R, " ***Blockchain for Enterprise Application Developers***", Wiley, 2020.
2.	Bina Ramamurthy "**Blockchain *in action***", Manning Publications, 2020.
3.	Kevin Werbach, ***"The Blockchain and the new architecture of trust"***, MIT Press, 2018.
4.	Joseph J. Bambara and Paul R. Allen, ***"Blockchain – A practical guide to developing business, law, and technology solutions"***, Tata McGraw-Hill Education, 2018.
	Joseph J. Bambara and Paul R. Allen, ***"Blockchain, IoT, and AI: Using the power of three to develop business, technical, and legal solutions"***, TataMcGraw-Hill Education 2019.
5.	Melanie Swan, ***"Blockchain – Blueprint for a new economy"***, OReilly publishers, 2015

Name of Program	M. Tech.	Semester – I	Year – I
Course– Name	**Bioinformatics**		
Course– Code	**ITME125**		
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**		
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**		
Course – Total Marks	**150**		
Course – Credits	**4**		
Course – Type	**Elective – 2**		

Prerequisites:

- Biostatistics and Probability
- Programming fundamentals
- Basic calculus and linear algebra

Course Outcomes:

Students will be able to –

CO-1	Understand fundamental concepts and terminologies in bioinformatics.
CO-2	Navigate and utilize various biological databases for data retrieval and analysis.
CO-3	Perform sequence alignment (pairwise and multiple) and analyze biological sequences for functional significance.
CO-4	Predict protein structures and interactions using computational tools and algorithms.
CO-5	Apply advanced computational techniques such as HMM and data mining in solving bioinformatics-related problems.

UNIT-1	**Introduction to Bioinformatics and Biological Databases:** Introduction to Bioinformatics: Overview, scope, and significance in biological research, Basic terminologies used in Bioinformatics Biological Databases: Databanks for nucleotides and proteins (GenBank, EMBL, DDBJ), Mapping, sequence, and structure databases, Non-redundant databases and their usage
UNIT-2	Biological Sequence Analysis Genome and Microarray Data Analysis, Pairwise Sequence Alignment: Local and global alignment, Dynamic programming methods: Needleman-Wunsch and Smith-Waterman algorithms, Gap penalties and scoring matrices (PAM, BLOSUM), Multiple Sequence Alignment (MSA): Progressive alignment (CLUSTAL W), Iterative alignment, Sum-of-pairs method and statistical significance of MSA, BLAST and motif searching

UNIT-3	**Structure Prediction in Proteins Protein Secondary and Tertiary Structure Prediction:** Methods: Homology modeling, threading, and ab initio, Protein folding and stability, Protein subcellular localization, Prediction Tools: OR Finder, PROSITE for motif detection, Structure prediction tools
UNIT-4	**Protein-Protein Interactions and Emerging Areas Protein-Protein Interaction Analysis:** Techniques for detecting and predicting interactions, Databases for protein interaction (e.g., STRING, BioGRID), Emerging Areas in Bioinformatics: Systems biology, Personalized medicine and bioinformatics, Synthetic biology
UNIT-5	**Computational Methods and Data Mining in Bioinformatics Computational Techniques for Sequence and Structure Analysis:** Hidden Markov Models (HMM), Pattern recognition and machine learning approaches, Soft Computing in Bioinformatics: Data mining methods applied to biological data, Use of neural networks and evolutionary algorithms, Applications of bioinformatics in drug discovery and genomics

Reference Books:

1.	Durbin, R., Eddy, S., Krogh, A. &Mitchison, G. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press.
2.	Jones, N.C. & Pevzner, P.A. (2004). An introduction to bioinformatics algorithms. MIT Press.
3.	Mount, D. W. (2001). Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press.
4.	Gibas, C. &Jambeck, P. (2001). Developing Bioinformatics Computer Skills. O'Reilly.
5.	Baxevanis, A. & Ouellette, B. F. F. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. Wiley-Interscience.

Name of Program	M. Tech.	Semester – I	Year – I

Course – Name	**Computational Geometry**
Course – Code	**ITME126**
Course – Periods / Week	**(L + T + P) ↔(3 + 1 + 0)**
Course – Exam Scheme	**(TA + FE + SE + ESE) ↔(20 + 15 + 15 + 100)**
Course – Total Marks	**150**
Course – Credits	**4**
Course – Type	**Elective – 2**

Course Contents:

UNIT-1	**Randomized Algorithms:** Quicksort, Randomized Quicksort, Expected Running Time Analyses, Quickselect, Randomized Quickselect, Expected, High Probability Bounds, Computational Geometric, Geometric Algorithms, Convex Hulls and Line Segment Intersection, Overlay of Planar Subdivisions and Polygon Triangulation, Delaunay Triangulations, Delaunay triangulations: divide-and-conquer, flip and incremental algorithms, duality of Voronoi diagrams, min-max angle properties.
UNIT-2	**Arrangements and Duality, Polytopes:** Guarding and Triangulations, Partitioning a Polygon into Monotone Pieces, Triangulating a Monotone Polygon, Linear Programming: Half-Plane Intersection, Incremental Linear Programming, Randomized Linear Programming, Unbounded Linear Programs, The Smallest Enclosing Disk Problem, Orthogonal Range Searching: 1-Dimensional Range Searching, Kd-Trees, Range Trees, Higher-Dimensional Range Trees, General Sets of Points, Fractional Cascading,
UNIT-3	**Point Location:** Point Location and Trapezoidal Maps, A Randomized Incremental Algorithm, Dealing with Degenerate Cases, A Tail Estimate, High Dimensional Convex Hulls and Voronoi Diagrams, Geometric Robustness, Binary Space Partitions and Applications, More Geometric Data Structures: Interval Trees, Priority Search Trees, Segment Trees, Binary Space Partitions: The Definition of BSP Trees, BSP Trees and the Painter's Algorithm, Constructing a BSP Tree, The Size of BSP Trees in 3-Space.
UNIT-4	**Arrangements of lines:** arrangements of hyperplanes, zone theorems, many-faces complexity and algorithms; Combinatorial geometry: Ham-sandwich cuts, Helly's theorems, k-sets, polytopes and hierarchies, polytopes and linear programming in d-dimensions, complexity of the union of convex sets, simply connected sets and visible regions.
UNIT-5	Motion Planning and Visibility Graphs, Meshing, Curve and Surface Reconstruction, Sweep techniques: plane sweep for segment intersections, Fortune's sweep for Voronoi diagrams, topological sweep for line arrangements; Randomization in computational geometry: algorithms, techniques for counting; Robust geometric computing; Applications of computational geometry, Range Searching, Clustering

	Point Sets using Quadtrees and Applications, Epsilon-Nets and VC Dimension, Shape Analysis and Shape Comparison.
Reference Books:	

1.	M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, ***Computational Geometry, Springer (2008).***
2.	F. P. Preparata and M. I. Shamos. ***Computational Geometry: An Introduction, Springer Verlag (1985)***.
3.	J. O' Rourke, ***Computational Geometry in C,*** Cambridge University Press (1998).
4.	H. Edelsbrunner, ***Algorithms in Combinatorial Geometry,pub: Springer-Verlag Berlin Heidelberg (1987)***
5.	K. Mulmuley, ***Computational Geometry: An Introduction Through Randomized Algorithms,*** Prentice Hall (1994).
6.	Michael J. Laszlo: ***Computational Geometry and Computer Graphics in C' Prentice Hall India (1996)***
7.	Mehlhorn and Naher, LEDA - ***a platform for combinatorial and geometric computing, pub: Cambridge (1999).***
	J. O'Rourke, ***Art Gallery Theorems and Algorithms,*** Oxford Univ. Press (1987).
8.	S.L. Devadoss and J. O'Rourke, ***Discrete and Computational Geometry, Princeton University Press (2011).***