Telephone No. 2419677/2419361
Fax: 0821-2419363/2419301

e-mail : registrar@uni-mysore.ac.in
www.uni-mysore.ac.in

UNIVERSITY OF MYSORE

Estd. 1916

VishwavidyanilayaKaryasoudha
Crawford Hall, Mysuru- 570 005

No.AC2(S)/55/2024-25

Dated: 20.07.2024

Notification

Sub:-Modification of Syllabus and Scheme of Examinations of Computer Science (PG) programmes from the Academic year 2024-25.

Ref:-1. Decision of Board of Studies in Computer Science (CB) meeting held on 15-06-2024.
2. Decision of the Faculty of Science & Technology meeting held on 19-06-2024.
3. Decision of the Academic Council meeting held on 28-06-2024.

*****

The Board of Studies in Computer Science (CB) which met on 15-06-2024 has resolved to recommend & approved the modified Syllabus and Scheme of examinations of Computer Science (PG) (M.Sc) programme with effect from the Academic year 2024-25.

The Faculty of Science & Technology and Academic Council at their meetings held on 19-06-2024 and 28-06-2024 respectively has also approved the above said modified Syllabus and Scheme of examinations hence it is hereby notified.

The Syllabus and Scheme of Examinations content may be downloaded from the University Website i.e., www.uni-mysore.ac.in.

Registrar

Registrar
University of Mysore
Mysore

To;

1. The Registrar (Evaluation), University of Mysore, Mysuru.
2. The Chairman, BOS/DOS in Computer Science, Manasagangothri, Mysore.
3. The Dean, Faculty of Science & Technology, DOS in Mathematics, MGM.
4. The Director, Distance Education Programme, Moulya Bhavan, Manasagangotri, Mysuru.
5. The Director, PMEB, Manasagangothri, Mysore.
6. Director, College Development Council,Manasagangothri, Mysore.
7. The Deputy Registrar/Assistant Registrar/Superintendent, Administrative Branch and Examination Branch, University of Mysore, Mysuru.
8. The PA to Vice-Chancellor/ Registrar/ Registrar (Evaluation), University of Mysore, Mysuru.
9. Office Copy.

ಮೈಸೂರು ವಿಶ್ವವಿದ್ಯಾನಿಲಯ

UNIVERSITY OF MYSORE

Department of Studies in Computer Science

Manasagangotri, Mysuru-570006

Regulations and Syllabus

Master of Computer Science

(M. Sc. [Two-year])

Under

Flexible Choice Based Credit System (FCBCS)

ಮೈಸೂರು ವಿಶ್ವವಿದ್ಯಾನಿಲಯ

UNIVERSITY OF MYSORE

GUIDELINES AND REGULATIONS LEADING TO MASTER OF COMPUTER SCIENCE

(M. Sc. TWO YEARS- SEMESTER SCHEME UNDER CBCS)

Programme Details:

Name of the Department	:	Department of Studies in Computer Science
Subject	:	Computer Science
Faculty	:	Science and Technology
Name of the Course	:	Master of Computer Science (M. Sc.)
Duration of the Course	:	M.Sc. 2 years- divided into 4 semesters

PROGRAMME OUTCOMES

- Graduates will acquire the knowledge about the current technology, trends, tools, theory of Computer Science and software development concepts to develop applications and to identify the potential problems where creative computer-based solutions can be applied to solve the problems.
- Graduates will be successful software professionals in IT industry capable of assimilating new information and understanding newer technology and its application domain to provide efficient and effective software solutions wherever possible.
- Graduates will inculcate the skills of communicating proficiently and collaborate successfully with peers, colleagues and organizations for higher studies, research and entrepreneurship to create new applications for the betterment of the society and their better future.

PROGRAMME SPECIFIC OUTCOMES

- Understand theories and application of emerging technologies.
- Expertise in computing enables students to solve complex, challenging problems.

PEDAGOGIES EMPLOYED IN THE M.SC., PROGRAMME

The pedagogy of teaching-learning involves three components.

- Lectures with intellectual inputs form the first component. This method provides Receptive Instructions to students.
- The second component is the tutorials. This method provides Directive Instructions to students.
- The third major component is the practical orientation with skills and participatory learning works. This method involves Exploratory Instructions.

M. Sc. (2 YEARS) IN COMPUTER SCIENCE

HARD CORE SUBJECTS

COURSE CODE	TITLE	CREDIT PATTERN (L: T: P)
MSCH1	Data Indexing Techniques	2:1:1
MSCH2	Theory of Languages	2:1:1
MSCH3	Distributed Databases	3:0:1
MSCH4	Object Oriented Programming with C++	3:0:1
MSCH5	Data Clustering Algorithms	3:0:1
MSCH6	Network Security	2:1:1
MSCH7	High Performance computing	3:0:1
MSCH8	Learning Models	2:1:1

SOFT CORE SUBJECTS

COURSE CODE	TITLE	CREDIT PATTERN (L: T: P)
MSCS1	Discrete Mathematical Structure	3:0:1
MSCS2	Algorithms	2:1:1
MSCS3	Fundamentals of Software Engineering	3:0:1
MSCS4	Advanced Software Engineering	3:0:1
MSCS5	Deep Learning	2:0:2
MSCS6	Internet of Things	3:0:1
MSCS7	Big Data Analytics	3:0:1
MSCS8	Artificial Intelligence	2:1:1
MSCS9	Medical Imaging	3:0:1
MSCS10	Digital Image Processing	3:0:1
MSCS11	.Net with C#	2:1:1
MSCS12	Sensor Networks	3:1:0
MSCS13	Data Mining and Data Warehousing	3:0:1
MSCS14	Cloud Computing	3:1:0
MSCS15	Mobile Communication	3:1:0
MSCS16	Probability and Statistics	3:1:0
MSCS17	E-Commerce and E- Governance	2:1:1
MSCS18	Information Retrieval	3:0:1
MSCS19	Java Programming	2:0:2
MSCS20	Compiler Construction	2:1:1
MSCS21	Computer Graphics	2:1:1
MSCS22	Object Oriented Design Patterns	2:1:1
MSCS23	Graph Theoretical Algorithms	3:1:0
MSCS24	Neural Networks	2:1:1
MSCS25	Fuzzy Sets and Theory	3:1:0
MSCS26	Soft Computing Techniques	2:1:1
MSCS27	Natural Language Processing	2:0:2
MSCS28	Essential Mathematics	3:1:0
MSCS29	Block-Chain Technology	2:1:1
MSCS30	Digital Signal Processing	2:1:1
MSCS31	Distributed Operating System	2:1:1
MSCS32	Linux Programming	2:1:1

Detailed Syllabus for M.Sc. Courses

HARD CORE

COURSE-I

DATA INDEXING TECHNIQUES

CREDIT (L: T: P)	2:1:1

Objectives:

- Learn the basics of data indexing and why it is important for efficient data retrieval.
- Explore different types of data indexing techniques, such as B-trees, hash indexing, and inverted indexing.
- Understand how to choose the right indexing technique based on the type of data and queries.
- Study the impact of indexing on database performance and storage requirements.
- Learn about the challenges and best practices in maintaining and updating indexes.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of data indexing.

CO2: Identify and explain various data indexing techniques and their applications.

CO3: Implement different data indexing structures such as B-trees, hash indexes, and bitmap indexes.

CO4: Evaluate the performance and efficiency of various indexing techniques in different database systems.

CO5: Apply appropriate indexing techniques to optimize query performance in real-world database scenarios.

Units	Course Content	Teaching Hours
Unit 1	Review on Binary tree, Search tree, AVL tree, operations, Balanced tree, B+ tree.	**10 Hours L(4):T(4):P(2)**
Unit 2	Multidimensional Data, Vector Data, Interval Data, KD Tree, R Tree	**10 Hours L(4):T(4):P(2)**
Unit 3	Strings, Indexing of Strings, 2-D Strings, Spatial Data Representation, 9DLT Matrix, Triangular Spatial Representation.	**10 Hours L(4):T(4):P(2)**
Unit 4	Hashing, Static Hashing, Dynamic Hashing, Operations, Applications	**10 Hours L(4):T(4):P(2)**

Textbooks:

1. “Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss.
3. "Algorithms in C++" by Robert Sedgewick.

Reference Books:

1. "Database Management Systems" by Raghu Ramakrishnan and Johannes Gehrke.
2. "Spatial Databases: A Tour" by Shashi Shekhar and Sanjay Chawla

COURSE-II

THEORY OF LANGUAGES

CREDIT (L: T: P) **2:1:1**

Objectives:

- ➢ Understand the fundamental concepts of formal languages and automata theory.
- ➢ Explore different types of formal languages, such as regular, context-free, and context-sensitive languages.
- ➢ Learn about various language-generating models, including finite automata, pushdown automata, and Turing machines.
- ➢ Study the applications of language theory in computer science, like compiler design and algorithm development.
- ➢ Recognize the limitations and capabilities of different language classes and models.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the foundational concepts and principles of formal languages and automata.

CO2: Describe and differentiate between various classes of languages, such as regular languages, context-free languages, and context-sensitive languages.

CO3: Construct and analyze different types of automata, including finite automata, pushdown automata, and Turing machines.

CO4: Apply formal grammar rules to generate and recognize strings in different formal languages.

CO5: Analyze and prove properties of languages and automata, such as decidability.

Units	Course Content	Teaching Hours
Unit 1	**Fundamentals of Formal Languages and Regular Languages:** Regular Languages and Regular Expressions, Finite Automata, Regular Grammars.	**10 Hours L(4):T(4):P(2)**
Unit 2	**Context-Free Languages and Pushdown Automata:** Pushdown Automata, Properties of Context-Free Languages, Context-Sensitive Grammars.	**10 Hours L(4):T(4):P(2)**
Unit 3	**Advanced Concepts in Formal Languages:** Properties of Turing Recognizable Languages, Chomsky Hierarchy, Formal Language Applications**.**	**10 Hours L(4):T(4):P(2)**
Unit 4	**Advanced Topics and Applications:** Automata and Formal Language Theory in Practice, Advanced Topics in Language Theory.	**10 Hours L(4):T(4):P(2)**

Textbooks:

1. Introduction to Automata Theory, Languages, and Computation by John E. Hopcroft, Rajeev
2. Motwani, Jeffrey D. Ullman, Pearson/Addison Wesley, 2007.

Reference Books:

1. An Introduction to Formal Languages and Automata by Peter Linz, Jones & Bartlett, Learning.

COURSE-III

DISTRIBUTED DATABASES

CREDIT (L: T: P)	3:0:1

Objectives:

- Learn the basics of distributed databases and how they differ from traditional databases.
- Understand the key concepts of data distribution, replication, and partitioning.
- Explore the challenges of distributed databases, such as consistency, concurrency, and fault tolerance.
- Study various architectures and models used in distributed databases.
- Learn about real world applications and benefits of distributed databases in handling large-scale data.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and architecture of distributed databases.

CO2: Design and implement distributed database systems, including data distribution and fragmentation.

CO3: Apply concurrency control and recovery techniques to ensure data integrity and consistency in distributed environments.

CO4: Evaluate the performance of distributed database systems and optimize queries for efficient data retrieval.

CO5: Address challenges related to distributed database security, including access control and data encryption.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Distributed Databases:** Fundamentals of Distributed Databases, Distributed Database Architecture, Distributed Data Independence, Distributed Database Design.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Distributed Database Models and Languages:** Distributed Database Models, Distributed Data Storage and Access, Distributed Database Languages, Distributed Transactions.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Distributed Database Management Systems (DDBMS):** Distributed DBMS Architecture, Data Replication and, Distributed Query Processing and Optimization, Distributed Data Security and Privacy.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Advanced Topics and Applications in Distributed Databases:** Big Data and Distributed Databases, Cloud-based Distributed Databases, NoSQL and NewSQL Databases, Future Trends and Research Directions.	**10 Hours L(8):T(8):P(2)**

Textbooks:

1. "Principles of Distributed Database Systems" by M. Tamer Özsu and Patrick Valduriez, Springer.
2. "Distributed Database Management Systems: A Practical Approach" by Saeed K. Rahimi and Frank S. Haug, Wiley.

Reference Books:

1. “Distributed Databases: Principles and Systems" by Stefano Ceri and Giuseppe Pelagatti, McGraw-Hill.
2. "Distributed and Cloud Computing: From Parallel Processing to the Internet of Things" by Kai Hwang, Jack Dongarra, and Geoffrey Fox, Morgan Kaufmann.
3. "Principles of Distributed Systems" by Vijay K. Garg, Springer.
4. "Advanced Database Systems" edited by Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard Snodgrass, V. S. Subrahmanian, and Roberto Zicari, Morgan Kaufmann

COURSE-IV

Object Oriented Programming with C++

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Learn the basics of C++ programming, including syntax and core concepts.
- Understand object-oriented programming principles like classes, inheritance, and polymorphism.
- Explore advanced features of C++ such as templates, exceptions, and the Standard Template Library (STL).
- Study memory management techniques, including pointers and dynamic allocation.
- Practice writing efficient and maintainable C++ code for various applications.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and syntax of the C++ programming language.

CO2: Write and debug C++ programs using basic programming constructs such as variables, control structures, functions, and arrays.

CO3: Implement object-oriented programming principles in C++, including classes, objects, inheritance, polymorphism, and encapsulation.

CO4: Utilize advanced features of C++, such as templates, exception handling, and the Standard Template Library (STL).

CO5: Develop efficient and robust C++ applications, and apply best practices for code organization, documentation, and testing.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** Concepts of Object-oriented programming, benefits of OOP, Applications of OOP, Structure of C++ program. Tokens, Keywords, Identifiers and constants, Data Types, Symbolic constants, Type compatibility, Declaration of variables, Dynamic initialization of variables, Reference variables, Operators in C++, Scope resolution operator, Member dereferencing operators, Memory management operators, Manipulators, Type cast operator, Expressions and their types, Special assignment expressions, Implicit conversions, Operator overloading.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Functions:** Function prototyping, Call by Reference, Return by Reference, Inline functions, Default arguments, const arguments, Function overloading, Friend and Virtual functions. **Classes and Objects:** Definitions, defining member functions, Making an Outside function Inline, Nesting of member functions, Private member functions, Arrays within a Class, Static data members, Static member functions, Arrays of Objects, Objects as function arguments, friendly functions, Returning Objects, const member functions. **Constructors and Destructors**: Types of Constructors, Constructors with default arguments, Dynamic initialization of objects, Constructing Two-dimensional arrays, Destructors.	**10 Hours L(8):T(0):P(2)**

Unit 3	**Operator Overloading and Type Conversions**: Defining operator overloading, overloading unary operators, Overloading Binary operators, Rules for overloading operators, Type conversions. **Inheritance:** Introduction, defining derived classes, Types of Inheritance, virtual base classes, abstract classes, constructors in derived classes, **Polymorphism:** Introduction, pointers, pointers to objects, this pointer, pointers to derived classes, virtual functions, pure virtual functions.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Files and Templates:** C++ streams, C++ stream classes, Unformatted I/O operations, Formatted I/O operations, Console I/O Operations, managing output with manipulators. **Files:** Classes for file stream operations, opening and closing a file, detecting end of file, more about **open():** file modes, file pointers and their manipulations, sequential input and output operations. **Templates:** Function templates, Class templates Exceptions.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Object Oriented Programming with C++ by E. Balagurusamy.
2. Object Oriented Programming in C++ by Robert Lafore Techmedia Publication.

Reference Books:

1. The complete reference C– by Herbert shieldt Tata McGraw Hill Publication.

COURSE-V

DATA CLUSTERING ALGORITHMS

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Learn about different data clustering algorithms and their purposes.
- Understand how to evaluate the effectiveness of clustering algorithms.
- Explore various techniques for improving clustering results.
- Study the applications of clustering algorithms in real-world scenarios.
- Recognize the challenges and limitations of clustering algorithms.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand and explain the fundamental concepts and principles of data clustering.

CO2: Identify appropriate clustering algorithms for different types of data and problem contexts.

CO3: Implement various clustering algorithms, such as k-means, hierarchical clustering, and DBSCAN.

CO4: Evaluate and compare the performance of different clustering algorithms using relevant metrics and validation techniques.

CO5: Apply data clustering techniques to real-world datasets and interpret the clustering results to derive meaningful insights.

Units	**Course Content**	**Teaching Hours**
Unit 1	**Data:** Data types, Data representation, pattern matrix, distance computing, proximity Matrix, similarity measures, dis-similarity matrix, data normalization techniques, min-max normalization, z-score normalization**.**	**10 Hours L(8):T(0):P(2)**
Unit 2	**Notion of data clustering:** Ambiguity in Data clusters, Taxonomy of data clustering, Agglomerative clustering approaches, single linkage, complete linkage, average linkage.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Partition clustering algorithms:** K- means clustering algorithm, K-Medoids clustering algorithm	**10 Hours L(8):T(0):P(2)**
Unit 4	**Cluster Validation:** confusion matrix, accuracy, precision, recall, f-score, sensitivity, specificity.	**10 Hours L(8):T(0):P(2)**

Reference Books:

1. Applied Multivariate Statistical Analysis by Johnson, R.A, and Dean W. Wichern.
2. An Introduction to Multivariate Analysis by Morrison D.
3. Multivariate Observations by Seber.
4. An Introduction to Multivariate Analysis by Anderson.
5. Analysis of Categorical data by Bishop.

COURSE-VI **NETWORK SECURITY**	
CREDIT (L: T: P)	**2:1:1**

Objectives:

- Understand the fundamentals of network security, including threats, vulnerabilities, and attacks.
- Explore common network security measures, such as firewalls, intrusion detection systems (IDS), and encryption.
- Learn about network security protocols like HTTPS, IPsec, and SSL/TLS.
- Study the importance of access control, authentication, and authorization in network security.
- Recognize emerging trends and challenges in network security, such as IoT security and cloud security.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of network security, including threats, vulnerabilities, and defense mechanisms.

CO2: Analyze network security requirements and design appropriate security architectures and policies.

CO3: Implement and configure security measures such as firewalls, intrusion detection systems (IDS), and virtual private networks (VPN).

CO4: Evaluate and assess network security risks and vulnerabilities using penetration testing and vulnerability assessment techniques.

CO5: Apply cryptographic techniques and protocols to secure network communications and data.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** computer security concepts, attacks, security services, security mechanisms; Classical encryption techniques-symmetric cipher models, substitution techniques, transposition techniques, rotor machines.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Symmetric ciphers:** Block cipher principles; DES Algorithm, strengths and weaknesses of DES, attacks on DES and defense, multiple encryptions; Asymmetric Ciphers Essential mathematics, public key cryptography.	**10 Hours L(6):T(2):P(2)**
Unit 3	RSA, Diffie Hellman key exchange, random number generation, Data integrity and authentication Hash functions; MAC; Digital signatures.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Key management**: Authentication, Web and system security, Web security; IP security; E mail security; System security-intruders, malicious software, firewalls.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Cryptography and Network Security-Principles and Practice, William Stallings, PEARSON.

Reference Books:

1. Cryptography and Network Security, Atul Kahate, Tata McGraw Hill.

COURSE-VII **HIGH PERFORMANCE COMPUTING**	
CREDIT (L: T: P)	**3:0:1**

Objectives:

- Learn about different computing technology architectures.
- Understand emerging trends in computing technology.
- Recognize the benefits of using advanced computing technologies.
- Explore how high-performance computing can solve complex problems.
- Study the impact of computing technologies on various industries.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand and articulate the fundamental principles of high-performance computing, including parallel processing and distributed systems.

CO2: Design and implement parallel algorithms to solve computationally intensive problems.

CO3: Utilize various parallel programming models and frameworks, such as MPI, OpenMP, and CUDA.

CO4: Analyze and optimize the performance of high-performance computing applications.

CO5: Apply high performance computing techniques to real-world problems, and evaluate their effectiveness and efficiency.

Units	Course Content	Teaching Hours
Unit 1	**Parallel Programming & Computing:** Introduction, Era of Computing, Parallel Computing, Multiprocessors and Multicomputer Architectures, Scalar VS Vector Processing, Multivector and Superscalar Machines, Pipelined Processors, SIMD Computers, Conditions of parallelism, Program flow mechanisms, Types of Parallelism – ILP, PLP, LLP, Program Partitioning and scheduling.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Introduction to High Performance Computing:** Era of Computing, Scalable Parallel Computer Architectures, towards low-cost computing, Network of Workstations project by Berkeley, Cluster Computing Architecture, Components, Cluster Middleware and SSI, Need of Resource Management and Scheduling, Programming Environments.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Cluster Computing:** Clustering Models, Clustering Architectures, Clustering Architectures key factors, types of clusters, Mission critical Vs Business Critical Applications, Fault Detection and Masking Algorithms, Check pointing, Heartbeats, Watchdog Timers, Fault recovery through Failover and Failback Concepts.	**10 Hours L(8):T(0):P(2)**
Unit 4	**High Speed Networks & Message Passing:** Introduction to High-Speed Networks, Lightweight Messaging Systems, Xpress Transport Protocol, Software RAID and Parallel File systems, Load Balancing Over Networks – Algorithms and Applications, Job Scheduling approaches and Resource Management in Cluster.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Rajkumar, High Performance Cluster Computing: Architectures and Systems, Vol. 1 Pearson Education.
2. Georg Hager and Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press.
3. Kai Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw Hill International Editions.

COURSE-VIII

LEARNING MODELS

CREDIT (L: T: P)	**2:1:1**

Objectives:

- Understand the fundamental concepts and theories of learning models.
- Analyze and evaluate the performance of different learning models.
- Implement various learning algorithms using appropriate tools and libraries.
- Apply learning models to practical problems in various domains.
- Critically assess the strengths and limitations of different learning models.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the theoretical foundations and principles of various learning models.

CO2: Identify and describe different types of learning models, including supervised, unsupervised, semi-supervised, and reinforcement learning.

CO3: Implement and apply machine learning algorithms and techniques to solve real-world problems and datasets.

CO4: Evaluate and compare the performance of different learning models using appropriate metrics and validation methods.

CO5: Explore advanced topics in learning models, such as deep learning architectures, transfer learning, and ensemble methods.

Units	Course Content	Teaching Hours
Unit 1	**Overview of Learning:** Machine Learning: Supervised, Unsupervised, Semi-supervised, Applications of Machine Learning, Features, Datasets, Stages in Machine Learning, Error analysis, Notion of proximity measures, validation measures for learning models, linear regression.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Classifier:** Nearest Neighbor Classifiers, Parametric and Non - Parametric classifier, Naïve Bayes Classifier, Decision Tree, support Vector Machines, Applications of Supervised Learning with Use Cases.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Curse of dimensionality**: dimensionality reduction, feature selection, filters and wrappers, Feature transformation techniques, Principal Component analysis, Linear Discriminant Analysis.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Neural Networks:** Introduction, Perceptron's, Multi-Layer Perceptron's, Backpropagation algorithm, Learning Error-Based Propagation in Neural Networks and Parameter Tuning.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. Richard O. Duda, Peter E. Hart, Pattern Classification, Wiley, 2000.
3. Yegnanarayana B., Artificial Neural Networks, Prentice-Hall of India Pvt.Ltd, 1999.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning, Springer, 2009.

Reference Books:

1. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, 2018.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2019 and Current Literature.

Detailed Syllabus for M.Sc. Courses

SOFT CORE

COURSE-I

DISCRETE MATHEMATICAL STRUCTURE

CREDIT (L: T: P)	3:0:1

Objectives:

- Explore foundational concepts in discrete mathematics, such as sets, relations, and functions.
- Study discrete structures like graphs, trees, and permutations.
- Understand combinatorial principles and counting techniques.
- Learn about logic, propositional and predicate calculus, and their applications in discrete mathematics.
- Recognize the relevance of discrete mathematical structures in computer science, cryptography, and algorithm design.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand and apply fundamental concepts in discrete mathematics, including sets, relations, functions, and logic.

CO2: Analyze and solve problems using principles from combinatorics and discrete probability.

CO3: Demonstrate proficiency in formal mathematical reasoning and proof techniques, such as mathematical induction and proof by contradiction.

CO4: Apply discrete mathematical structures to computer science and engineering applications, such as algorithms, data structures, and cryptography.

CO5: Explore advanced topics in discrete mathematics, such as graph theory, automata theory, and formal languages.

Units	Course Content	Teaching Hours
Unit 1	**Sets:** Set basics, Venn diagrams, Union, intersection, set difference, complement, Cartesian product, Power sets, Cardinality of finite sets. **Relation:** Reflexivity, symmetry, antisymmetric, transitivity, Equivalence relations, partial orders. **Function:** Domain, target, and range/image of a function, surjection, injections, bijections, inverses, composition.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Basic Logic:** Propositional logic, Logical connectives, Truth tables, Disjunctive normal form, Validity of a well-formed formula, Propositional inference rules, Universal and existential quantifiers and their negations. Proof Techniques: Proof by Induction. **Group theory**: Groups, subgroups, generators and evaluation of powers, cosets and Lagrange's theorem, permutation groups and Burnside's theorem, isomorphism, automorphisms, homomorphism, monoids, concepts of rings, fields. Introduction to vector space.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Counting:** The basics of counting, the pigeonhole principle, permutations and combinations, recurrence relations, solving recurrence relations, generating functions, inclusion-exclusion principle and application of inclusion-exclusion, Basic modular arithmetic.	**10 Hours L(8):T(0):P(2)**

Unit 4	**Discrete Probability:** Finite probability space, events, Properties of events, Conditional probability, Bayes‟ theorem, Independence. Statistical Distribution: Discrete Distribution, Binomial distribution, Gamma distribution, Beta distribution, Chi-square distribution, Univariate normal distribution.	**10 Hours** **L(8):T(0):P(2)**

Textbooks:

1. Edgar Goodaire and Michael Parmenter, Discrete Mathematics with Graph Theory, Third Edition, PHI, ISBN-13-9750131679955.
2. S. Lipschutz, Discreate Mathematics, TMH, ISBN 0-07-066932-0.

Reference Books:

1. Bernard Kolman C,Busby and Sharon Ross, Discrete Mathematical Structures, 2007, ISBN - 81-203-2082-4, Publication PHI.

COURSE-II

ALGORITHMS

CREDIT (L: T: P)	2:1:1

Objectives:

- Understand the basics of algorithms, which are step-by-step procedures or formulas for solving problems.
- Explore different types of algorithms: sorting, searching, graph traversal, and dynamic programming.
- Study algorithm design techniques such as divide and conquer, greedy algorithms, and backtracking.
- Learn about algorithm analysis, including time complexity (Big O notation).
- Recognize the importance of algorithms in computer science, data analysis, and optimization problems.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand and analyze the fundamental principles and techniques of algorithms.

CO2: Implement and evaluate basic algorithms for sorting, searching, and data structures.

CO3: Design and analyze algorithm efficiency using asymptotic notation (Big-O notation) and complexity analysis.

CO4: Apply algorithmic techniques such as divide and conquer, dynamic programming, and greedy algorithms to solve computational problems.

CO5: Solve real-world problems by selecting and adapting appropriate algorithms.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Algorithm**: Algorithm Specification, Analysis Framework, Performance Analysis: Space complexity, Time complexity. Asymptotic Notations: Mathematical analysis of non-recursive and recursive Algorithms with Examples. Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. **Fundamental Data Structures**: Stacks, Queues, Graphs, Trees, Sets and Dictionaries.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Divide and Conquer:** General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum, Merge sort, Quick sort, Strassen‟s matrix multiplication, Advantages and Disadvantages of divide and conquer, **Decrease and Conquer Approach**: Topological Sort.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Greedy Method:** General method, Coin Change Problem, Knapsack Problem, Minimum cost spanning trees, Prim‟s Algorithm, Kruskal‟s Algorithm, Dijkstra's Algorithm, Optimal Tree problem: Huffman Trees and Codes. Transform and Conquer Approach: Heaps and Heap Sort. **Dynamic Programming:** General method with Examples, Multistage Graphs. **Transitive Closure:** Warshall‟s Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem, Bellman-Ford Algorithm, Travelling Sales Person problem, Reliability design.	**10 Hours L(6):T(2):P(2)**

Unit 4	**Backtracking:** General method, N-Queens problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles. **Branch and Bound**: Assignment Problem, Travelling Sales Person problem, 0/1. **Knapsack problem:** LC Branch and Bound solution, FIFO Branch and Bound solution. NP-Complete and NP-Hard problems: Basic concepts, nondeterministic algorithms, P, NP, NP-Complete, and NP-Hard classes.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 2rd Edition, 2009. Pearson.
2. Ellis Horowitz, Satraj Sahni and Rajasekaran, Computer Algorithms/C++, 2nd Edition, 2014, Universities Press.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, Introduction to Algorithms, 3rd Edition, PHI.
2. S. Sridhar, Design and Analysis of Algorithms, Oxford (Higher Education).

COURSE-III

FUNDAMENTALS OF SOFTWARE ENGINEERING

CREDIT (L: T: P)	3:0:1

Objectives:

- Learn the steps and challenges in developing software, and understand the importance of following a structured process.
- Understand how to gather information, analyze requirements, and plan projects, including estimating, scheduling, managing risks, and handling changes.
- Know the key concepts and best practices in software design, coding standards, and testing methods.
- Learn about software reliability, quality metrics, and maintenance.
- Understand how Computer-Aided Software Engineering (CASE) tools support software development.

Course outcomes:

At the end of the course the students will be able to

CO1: An overview of typical software development process activities.

CO2: Guide the software development team or work in a team very efficiently and effectively.

CO3: Practical knowledge about the coding platform, test environment, debugging process and software maintenance.

CO4: Knowledge about estimating project parameters, estimating techniques and risk management activities.

CO5: Knowledge of assessing the feasibility of the software project to be under taken.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** The role of software engineering in system design, software products, emergence of software engineering, notable changes in software development practices, the changing nature of software, the software engineering challenges, Software processes, desired characteristics of software process, the software life cycle, software development process models, comparison of process models. **Requirement analysis and specification**: need for SRS, characteristics of SRS, organization of SRS document. Techniques for representing complex logic, functional specification with Use Cases, formal system development techniques.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Introduction to software design**: characteristics of a good software design, module level concepts, and software design approaches. **Function-oriented software design**: Overview of the structured analysis and structured design methodology, data flow diagrams, structured design. **Object-oriented software design concepts**: Overview, UML, object-oriented design methodology, OOD metrics and goodness criteria. **User-interface design**: Characteristics, basic concepts, types of user interfaces, and a GUI design methodology	**10 Hours L(8):T(0):P(2)**
Unit 3	**Introduction to coding**: coding standards, guidelines, code walkthroughs, code inspections, software documentation, unit testing, black box testing, white box testing, integration testing, system testing, general issues associated with testing, debugging, approaches and guidelines, **program analysis tools**: Software reliability metrics, software reliability specification, software quality factors, quality metrics, software quality management system. The maintenance process, software reverse engineering, software maintenance process models, estimation of maintenance costs, CASE and its scope, CASE support in software life cycle, characteristics of CASE tools, architecture of CASE environment	**10 Hours L(8):T(0):P(2)**

Unit 4	**Introduction to software project management**: responsibility of software project managers, project planning, structure of software project management document, Project size estimation metrics, project estimation techniques, project scheduling and staffing, Gantt charts, PERT charts, organization and team structures, attributes of a good software engineer, risk management and configuration management.	**10 Hours** **L(8):T(0):P(2)**

Textbooks:

1. Software Engineering by Ian Sommerville – Fifth Edition, Addison-Wesley.
2. An Integrated Approach to Software Engineering by Pankaj Jalote – Third Edition.
3. Fundamentals of Software Engineering by Rajib Mall – PHI.
4. Software Engineering by Roger S. Pressman – Sixth Edition, Mc Graw Hill.
5. Fundamentals of Software Engineering by Ghezzi, Jazayeri, Mandrioli – PHI.

COURSE-IV

ADVANCED SOFTWARE ENGINEERING

CREDIT (L: T: P)	**3:0:1**

Objectives:

- To introduce the idea of critical system, to understand how to specify functional and non-functional requirements for critical systems, to introduce implementation techniques used in the development of critical systems and to discuss verification and validation techniques used in the development of critical systems.
- To introduce the concepts of software architecture and architecture design, to discuss models of the software architecture for distributed systems, to introduce architectural models for specific classes of application software systems, to introduce techniques used in the design of real–time systems and to describe some generic real-time system architectures.
- To describe the approaches to rapid software development, to introduce software reuse and explain how it contributes to the software development process, to describe a software development process based on the composition of reusable, standardized components.
- To introduce software quality management and software measurement, to explain how software process can be improved to produce better software, to introduce issues to be considered in the specification and design of secure software, to introduce service-oriented software engineering and aspect-oriented software development.

Course outcomes:

At the end of the course the students will be able to

CO1: Acquire the knowledge of developing software applications for critical systems and its validation.

CO2: Able to propose architecture design for distributed systems, real-time systems and their development.

CO3: Explore the feasibility of rapid software development techniques and the ideas of software reuse to accelerate the software development process.

CO4: Utilize advanced tools and frameworks for software development, testing, and deployment.

CO5: Able to assess the quality of software process to produce better software, secure software and also service-oriented and aspect-oriented software development.

Units	Course Content	Teaching Hours
Unit 1	**Socio-technical systems:** Introduction, emergent system properties, system engineering, organization, people and computer systems, legacy systems. **Critical Systems**: Introduction, a safety-critical system, system dependability, availability and reliability, safety, security. **Critical systems specification:** Risk driven specification, safety specification, security specification, software reliability specification. **Critical systems development:** Dependable processes, dependable programming, fault tolerance, fault-tolerance architectures. Critical system validation: Reliability validation, safety assurance, security assessment, safety and dependability cases.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Distributed systems architectures:** Introduction, multiprocessor architectures, client-server architectures, distributed object architectures, inter-organizational distributed computing. **Application architectures**: Data processing systems, transaction processing systems, event processing systems, language processing systems. **Real-time software design**: System design, real-time operating systems, monitoring and control systems, data acquisition systems.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Rapid software development:** Agile methods, extreme programming, rapid application development, software prototyping. **Software reuse:** The reuse landscape, design patterns, generator-based reuse, application frameworks, application system recuse. **Component-based software engineering:** Components and component models, the CBSE process, component decomposition.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Quality management:** Process and product quality, quality assurance and standards, quality planning, quality control, software measurement and	**10 Hours L(8):T(0):P(2)**

	metrics. **Process improvement**: Process classification, process measurement, process analysis and modeling, process change, the CMMI process improvement framework. **Security engineering**: security concepts, security risk management, design for security, system survivability. **Service-oriented software engineering:** Service as reusable components, service engineering, software development with services. **Aspect- oriented software development:** The separation of concerns, aspects, join points and point cuts, software engineering with aspects.	

Textbooks:

1. "Software Engineering: A Practitioner's Approach" by Roger S. Pressman and Bruce Maxim, McGraw-Hill Education.
2. "Software Engineering: Modern Approaches" by Eric J. Braude and Michael E. Bernstein, Wiley.
3. "Advanced Software Engineering: Expanding the Frontiers of Software Technology" by Hakan Erdogmus, O'Reilly Media.
4. "Software Engineering: Theory and Practice" by Shari Lawrence Pfleeger and Joanne M. Atlee, Pearson.
5. "Advanced Software Testing - Vol. 1, 2nd Edition: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst" by Rex Black

COURSE-V

DEEP LEARNING

CREDIT (L: T: P)	2:0:2

Objectives:

- Understand the fundamentals of deep learning, a subset of machine learning focused on neural networks.
- Explore various types of neural networks used in deep learning, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
- Learn about deep learning frameworks like TensorFlow and PyTorch used for implementing and training neural networks.
- Study applications of deep learning in areas such as computer vision, natural language processing, and speech recognition.
- Recognize challenges and advancements in deep learning, including model interpretability, scalability, and ethical considerations.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the foundational concepts and principles of deep learning.

CO2: Implement and train deep neural networks using frameworks such as TensorFlow or PyTorch.

CO3: Apply various deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs).

CO4: Evaluate and optimize deep learning models using techniques such as regularization, dropout, and hyperparameter tuning.

CO5: Apply deep learning to solve real-world problems in areas such as computer vision, natural language processing, and reinforcement learning.

Units	Course Content	Teaching Hours
Unit 1	**Deep Feed forward Networks**: Gradient-Based Learning, Hidden Units, Architecture Design, Back Propagation. **Regularization:** Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi Supervised Learning, Multi-Task Learning.	**10 Hours L(5):T(0):P(5)**
Unit 2	**Optimization for Training Deep Models**: How Learning Differs from Pure Optimization, Challenges in Neural Network Optimization, Basic Algorithms. Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates. **Convolutional Networks**: The Convolution Operation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Variants of the Basic Convolution Function, Efficient Convolution Algorithms, Random or Unsupervised Features.	**10 Hours L(5):T(0):P(5)**
Unit 3	**Sequence Modelling**: Recurrent and Recursive Nets, Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Encoder-Decoder Sequence-to-Sequence Architectures, Deep Recurrent Networks, Recursive Neural Networks. Long Short-Term Memory (LSTM).	**10 Hours L(5):T(0):P(5)**
Unit 4	**Introduction to Reinforcement Learning:** State of the art applications in Atari, Alpha Go, relation to other problems in artificial intelligence, **Markov Decision Processes (model based):** Formulation, Value Iteration (VI), Policy Iteration (PI), Linear Programming, Approximate	**10 Hours L(5):T(0):P(5)**

Textbooks:

1. Deep Learning, Lan Good fellow and Yoshua Bengio and Aaron Courville, MIT Presshttps://www.deeplearn ingbook.org/, 2016.
2. Richard S. Sutton and Andrew G. Barto, Introduction to Reinforcement Learning, 2nd Edition, MIT Press. 2017. ISBN-13 978-0262039246.

3. Dimitri Bertsekas and John G. Tsitsiklis, Neuro Dynamic Programming, Athena Scientific. 1996. ISBN-13: 978-1886529106.

Reference Books:

1. Neural Networks, A systematic Introduction, Raúl Rojas, 1996.
2. Pattern Recognition and machine Learning, Chirstopher Bishop, Springer, 2007.
3. V. S. Borkar, Stochastic Approximation: A Dynamical Systems Viewpoint, Hindustan Book Agency, 2009. ISBN-13: 978-0521515924.
4. Deep Learning. Ian Goodfellow and Yoshua Bengio and Aaron Courville. MIT Press. 2016.ISBN-13: 978-0262035613.

COURSE-VI

INTERNET OF THINGS

CREDIT (L: T: P)	3:0:1

Objectives:

- Understand the concept of the Internet of Things (IoT) and its role in connecting everyday objects to the internet.
- Explore IoT architectures, including sensors, actuators, communication protocols, and cloud computing.
- Learn about IoT applications across various industries, such as smart homes, healthcare, agriculture, and industrial automation.
- Study IoT security considerations, including data privacy, authentication, and network integrity.
- Recognize the challenges and future trends in IoT, such as interoperability, scalability, and edge computing.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and components of the Internet of Things (IoT), including sensors, actuators, and IoT platforms.

CO2: Design and develop IoT systems and applications using microcontrollers, sensors, and communication protocols.

CO3: Implement IoT communication protocols such as MQTT, CoAP, and HTTP(S).

CO4: Analyze and interpret IoT data streams using cloud-based IoT platforms and edge computing techniques.

CO5: Evaluate security and privacy issues in IoT systems and implement appropriate measures for securing IoT deployments.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to IoT:** Definition and Characteristics, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies. **M2M and IoT:** Introduction to M2M, Difference between IoT and M2M, SDN and NFV for IoT. **IoT Protocols:** IEEE 802.15.4, BACNet Protocol, Modbus, KNX, Zigbee Architecture, 6LoWPAN, RPL.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Developing Internet of Things:** IoT Platforms Design Methodology, Python packages of Interest for IoT, IoT Physical Devices and Endpoints **IoT and Cloud**: IoT Physical Servers and Cloud Offerings, **IoT Tools:** Chef, Puppet.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Data Analytics for IoT:** Big Data Platforms for the IoT, Hadoop Map Reduce for Batch Data Analysis, Apache Oozie Workflows for IoT Data Analysis, In-Memory Analytics using Apache Spark, Apache Storm for Real Time Data Analysis, Sustainability Data and Analytics in Cloud based M2M Systems, **Fog Computing**: A Platform for IoT and Analytics.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Domain Specific IoTs:** Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health and Lifestyle, Virtual Reality Internet Advertising, Intelligent Transportation Systems, **Health Information System:** Genomics Driven Wellness Tracking and Management System (Go-WELL).	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-on Approach,2015,1st Edition, Universities Press.
2. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things–Key applications and Protocols,2012, Wiley Publication.
3. Honbo Zhou, The Internet of Things in the Cloud: A Middleware Perspective,2012, CRC Press.

Reference Books:

1. Dieter Uckelmann; Mark Harrison; Florian Michahelles Architecting the Internet of Things,2011, Springer.

2. Internet of Things: Architecture and Design Principles-Raj Kamal,1stEdition, McGraw Hill Education.
3. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things-David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry,1st Edition, Pearson Education.
4. Internet of Things-Srinivasa K G, CENGAGE Leaning India.

COURSE-VII

BIG DATA ANALYTICS

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Understand the concept of big data analytics, which involves processing and analyzing large volumes of data to uncover patterns and insights.
- Explore big data technologies and frameworks like Hadoop, Spark, and Kafka used for storing, processing, and analyzing massive datasets.
- Learn about data mining techniques, machine learning algorithms, and statistical methods applied in big data analytics.
- Study real-time analytics and predictive analytics capabilities enabled by big data technologies.
- Recognize the impact of big data analytics in various industries, such as healthcare, finance, marketing, and transportation, for making data-driven decisions and improving business outcomes.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of big data analytics, including the characteristics of big data and its challenges.

CO2: Analyze and pre-process large-scale datasets using distributed computing frameworks such as Hadoop and Spark.

CO3: Implement and apply various big data analytics techniques, including data mining, machine learning, and statistical analysis.

CO4: Design and develop scalable data processing pipelines for big data applications.

CO5: Evaluate the performance and efficiency of big data analytics solutions and optimize them for specific use cases and data volumes.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Big Data Analytics:** Big Data Overview, State of practice in analytics, Role of Data Scientists, Examples of Big Data Analytics, Data Analytics Lifecycle, Components of Hadoop, Analyzing Big data with Hadoop, Design of HDFS, developing a Map reduce Application.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Map Reduce**: Distributed File System (DFS), Map Reduce, Algorithms using Map Reduce, Communication cost Model, Graph Model for Map Reduce Problem.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Hadoop Environment:** Setting up a Hadoop Cluster, Hadoop Configuration, Security in Hadoop, Administering Hadoop, Hadoop Benchmarks, Hadoop in the cloud. **Big Data Analytics Methods using R**: Introduction to R-Attributes, R Graphical user interfaces, Data import and export, attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Statistical methods for evaluation:** Hypothesis Testing, Difference of Means, Wilcoxon Rank-Sum Test, Type I and Type II errors, power and sample size, ANOVA. Advanced Analytics - technologies and tools: Analytics for unstructured data, The Hadoop ecosystem – pig – Hive-HBase- Mahout- NoSQL.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data by EMC Education Services, 2015, publishing.
2. Anand Raja Raman and Jeffrey David Ullman, Mining of Massive Datasets, 2012, Cambridge University Press.

Reference Books:

1. Tom White, Hadoop: The Definitive Guide, 3rd Edition, O'Reilly Media.
2. Data mining and Data warehousing by Parteek Bhatia.

COURSE-VIII

ARTIFICIAL INTELLIGENCE

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Understand the field of artificial intelligence (AI) and its goal of creating intelligent machines capable of performing tasks that typically require human intelligence.
- Explore different branches of AI, including machine learning, natural language processing, computer vision, and robotics.
- Learn about AI techniques such as supervised learning, unsupervised learning, reinforcement learning, and neural networks.
- Study AI applications across industries, such as healthcare, finance, autonomous vehicles, and personalized recommendations.
- Recognize ethical considerations and societal impacts of AI, including bias in algorithms, job displacement, and privacy concerns.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the foundational concepts and principles of artificial intelligence.

CO2: Implement and apply AI algorithms and techniques, including search algorithms, knowledge representation, and reasoning.

CO3: Develop AI systems capable of learning from data using machine learning algorithms such as supervised, unsupervised, and reinforcement learning.

CO4: Apply natural language processing (NLP) techniques for text analysis and generation.

CO5: Evaluate ethical and societal implications of AI technologies and apply responsible AI practices.

Units	Course Content	Teaching Hours
Unit 1	**Foundations and Problem-Solving in AI**: Introduction to AI: History of Artificial Intelligence, State of the Art Artificial Intelligence, Risks and Benefits of AI. Intelligent Agents: Agents and Environments, Nature of Environments, Structure of Agents. **Problem-Solving by Searching:** Problem-Solving Agents, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Advanced Search Techniques and Game Playing**: Search in Complex Environments, Local Search and Optimization Problems, Local Search in Continuous Spaces, Search with Nondeterministic Actions. **Adversarial Search and Games:** Game Theory, Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search and Limitations of Game Search Algorithms.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Knowledge Representation and Reasoning**: Logical Agents: Knowledge Based Agents, Propositional Logic, Agents Based on Propositional Logic, Predicate Logic, Knowledge Representation: Categories and Objects, Events, Reasoning Systems for Categories, Reasoning.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Planning, Uncertainty, and Learning:** Planning, Definition of Classical Planning, Algorithms for Classical Planning, Heuristics for Planning, Hierarchical Planning, Uncertain Knowledge and Reasoning: Bayes' Rule and Its Use, Naive Bayes Models, Probabilistic Reasoning. **Expert Systems and Learning**: Rote Learning, Learning by Advice, Learning by Analogy, Macro Learning.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Russell, S., &Norvig, P. (2021). Artificial Intelligence: A Modern approach (4th ed.). Pearson.
2. Nilsson, N. J. (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann.
3. Rich, E., & Knight, K. (1991). Artificial Intelligence. McGraw-Hill.
4. Luger, G. F. (2008). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th ed.). Addison-Wesley.
5. Poole, D., &Mackworth, A. (2010). Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press.

Reference Books:

1. Russell, S., &Norvig, P. (2021). Artificial Intelligence: A Modern approach(4th ed.). Pearson.

COURSE-IX

MEDICAL IMAGING

CREDIT (L: T: P)	**2:0:2**

Objectives:

- Understand the principles behind medical imaging technologies like MRI, CT scans, and X-rays.
- Explore the applications of medical imaging in diagnosing diseases and injuries.
- Learn about the advancements in medical imaging technology, such as 3D imaging and AI-assisted diagnosis.
- Understand the importance of image quality, resolution, and safety in medical imaging.
- Learn about transformation techniques for image enhancement and restoration.

Course outcomes:

At the end of the course the students will be able to

CO1: Medical imaging modalities such as projection radiography, computed tomography, nuclear medicine, ultrasound imaging, magnetic resonance imaging, etc.

CO2: Basic principles of Optics and characteristics of X-ray, CT, MRI, images along with an understanding of the biological effects of ionizing radiation.

CO3: Study image identification Classification and segmentation methods in in Medical images using Machine learning and Deep Learning concepts.

CO4: Familiarize yourself with popular open-source tools for image processing tools used in Medical Imaging.

CO5: Apply medical imaging techniques to clinical scenarios and research, interpreting the results to assist in diagnosis and treatment planning.

Units	**Course Content**	**Teaching Hours**
Unit 1	**Optics for Medical imaging:** Reflection and Refraction, Mirrors, Prisms, Convex, Concave Lenses, Image Formation, Pinhole Camera, Refraction by Curved Surfaces, Focal length, Focusing and Power. **Electromagnetic Waves:** Wave Propagation, Amplitude, Intensity, Relationship between Frequency and Wavelength. Electromagnetic waves, Sources of Electromagnetic Radiation, Electromagnetic Spectrum, Wave-particle duality - Photons, Energy of Photons.	**10 Hours L(5):T(0):P(5)**
Unit 2	**Optical Imaging:** Polarization, Coherence and Interference, Fourier Optics: 2-D and 3-D Imaging Systems, Holographic Imaging Systems. Principles of Image Detectors for Visible and Infrared, Image storage Technologies, Adaptive Optical Imaging Systems. **Imaging Modalities in Medical Science:** Projection radiography, Computed Tomography, Nuclear medicine, Ultrasound imaging, Magnetic Resonance Imaging, Fluoroscopy, Angiography, Digital radiography, Mammography.	**10 Hours L(5):T(0):P(5)**
Unit 3	**X-ray Imaging:** Production and Properties of X-rays, X-ray Tube Design X-Ray Circuits including Components and Control, Imaging. **Computed Tomography (CT):** Principles of CT Scanning, Image Reconstruction and Display, **Virtual Reality Imaging. Mammography:** Principle, Application, Advantage in Soft Tissue Radiography. **Magnetic Resonance Imaging (MRI):** Basic Ideas, Magnetic Susceptibility, Nuclear Magnetic Moments, External Magnetic Field, Radio Frequency (RF). **Ultrasound Imaging:** Basic Ideas, Interaction with Tissues, Physics of Ultrasound Propagation, Transducers. **Nuclear Imaging:** Imaging Techniques and Scan, Radioactive Tracers, PET-CT.	**10 Hours L(5):T(0):P(5)**
Unit 4	**Study of Open-Source Tools for image processing**: Image J and FIJI, OSIRIX, 3D Slicer, FSL. **Case Study:** Experimentation of Machine learning and Deep learning algorithm such identification, classification and Segmentation on Different image acquisition techniques of Medical imaging.	**10 Hours L(5):T(0):P(5)**

Reference Books:

1. Fundamentals of Medical Imaging, Paul Suetens, Cambridge University Press.
2. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods, Jiri Jan, CRC Press.
3. Medical image processing: the mathematics of medical imaging, James A. Green, Greenwood Research.
4. Handbook of Medical Image Processing and Analysis, Isaac Bankman, Academic Press
5. J. W. Goodman, Fourier Optics, 3rd Edition, Roberts & Co., 2005
6. Yariv, Optical Electronics in Modern Communications, 5th Edition, Oxford University Press, 1997.
7. B. Saleh and M. Teich, Fundamentals of Photonics, Wiley, 1991.
8. J. G. Brown, X-Rays and Their Applications, Springer New York,NY, 2011.
9. Karuppasamy Subburaj, CT Scanning - Techniques and Applications, Ebook (PDF), ISBN 978-953-51-6768-6, 2011
10. Hussey, Matthew, Basic physics and technology of medical diagnostic ultrasound, London: Macmillan, 1985
11. Haim Azhari, Basics of Biomedical Ultrasound for Engineers, John Wiley & Sons, Inc.
12. 2010
13. Brian M. Dale, Mark A. Brown and Richard C. Semelka, MRI Basic
14. Principles and Applications, John Wiley & Sons, Ltd, 2015
15. Fred A Mettler and Milton J Guiberteau, Essentials of Nuclear Medicine Imaging, Elsevier Inc. 2012.

COURSE-X

DIGITAL IMAGE PROCESSING

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Understand the fundamentals of digital image processing, which involves manipulating digital images using computer algorithms.
- Explore basic image processing techniques such as image enhancement, restoration, and segmentation.
- Learn about advanced image processing methods like image compression, feature extraction, and object recognition.
- Study the applications of digital image processing in fields such as medical imaging, satellite imagery analysis, and multimedia content creation.
- Recognize the challenges in digital image processing, including noise reduction, image quality improvement, and real-time processing requirements.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the fundamental concepts and principles of digital image processing.

CO2: Apply image enhancement techniques to improve the quality of digital images.

CO3: Implement image segmentation algorithms to partition images into meaningful regions.

CO4: Develop image restoration methods to remove noise and artifacts from images.

CO5: Utilize feature extraction and pattern recognition techniques for object detection and recognition in images.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Digital Image Processing:** Course overview, History and applications of image processing, Basic concepts and definitions. **Digital Image Fundamentals:** Image representation and models, Sampling and quantization, Color models (RGB, CMY, HSV). **Image Enhancement in the Spatial Domain:** Point processing techniques, Histogram processing, Spatial filtering (smoothing and sharpening).	**10 Hours L(8):T(0):P(2)**
Unit 2	**Image Enhancement in the Frequency Domain:** Fourier transform and its properties, Frequency domain filtering, Image smoothing and sharpening in the frequency domain. **Image Restoration:** Degradation model, Noise reduction techniques, Inverse filtering and Wiener filtering. **Color Image Processing:** Color transformation, Color enhancement techniques, Color segmentation.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Wavelets and Multiresolution Processing:** Introduction to wavelets, Multiresolution analysis, Applications of wavelets in image processing. **Image Compression:** Fundamentals of image compression, Lossless and lossy compression techniques, JPEG and JPEG2000 standards. **Morphological Image Processing:** Basic morphological operations, Morphological algorithms, Applications in shape analysis.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Image Segmentation:** Edge detection techniques, Region-based segmentation, Segmentation using clustering methods. **Representation and Description:** Boundary representation, regional descriptors, Shape representation and description. **Object Recognition:** Pattern recognition fundamentals, Feature extraction and selection, Machine learning for object recognition. **Advanced Topics and Applications:** Introduction to deep learning in image processing, medical image processing.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, 4th Edition, Pearson.
2. "Digital Image Processing Using MATLAB" by Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins.

Reference Books:

1. "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, and Roger Boyle.

COURSE-XI

.NET WITH C#

CREDIT (L: T: P) **2:1:1**

Objectives:

- Understand the fundamentals of the .NET framework, including its architecture, CLR (Common Language Runtime), and class libraries.
- Learn the basics of C# programming, including syntax, data types, control structures, and object-oriented principles like encapsulation, inheritance, and polymorphism.
- Explore advanced C# features such as delegates, events, LINQ (Language Integrated Query), and asynchronous programming with async/await.
- Study the development of desktop applications using Windows Forms and WPF (Windows Presentation Foundation), as well as web applications using ASP.NET.
- Examine best practices for .NET application development, including debugging, exception handling, performance optimization, and secure coding practices.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the fundamentals of the C# programming language, including syntax, data types, control structures, and object-oriented programming principles.

CO2: Develop console applications, graphical user interfaces (GUIs), and web applications using C# and the .NET framework.

CO3: Implement and utilize .NET features such as LINQ (Language Integrated Query), asynchronous programming, and parallel programming.

CO4: Design and develop database applications using ADO.NET or Entity Framework for data access.

CO5: Apply best practices for software development, including debugging, testing, and deploying C#/.NET applications.

Units	Course Content	Teaching Hours
Unit 1	**Review of OOP Concepts:** Overview of .NET Framework, Basic Elements of C#, Program Structure and simple Input and Output Operations, Operators and Expressions, Statements, Arrays and Structures.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Inheritance:** Namespace, Polymorphism, Interface and Overloading, Multiple Inheritance, Property, Indexes, Delegates, Publish/Subscribe Design Patterns, Operator Overloading Method Overloading	**10 Hours L(6):T(2):P(2)**
Unit 3	**C# Concepts for creating Data Structures:** File Operation, File Management systems, Stream Oriented Operations, Multitasking, Multithreading, Thread Operation, Synchronization.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Working with XML:** Techniques for Reading and Writing XML Data, Using XPath and Search XML, ADO.NET Architecture, ADO.NET Connected and Disconnected Models, XML and ADO.NET, Simple and Complex Data Binding, Data Grid View Class.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. S. Thamarai Selvi and R. Murugesan “A Textbook on C# “, Pearson Education, 2003.
2. Stephen C. Perry “Core C# and .NET”, Pearson Education, 2006.

Reference Books:

1. Jesse Liberty, “Programming C#”, Second Edition, O’Reilly Press,2002.
2. Robinson et al, “Professional C#”, Fifth Edition, Wrox Press,2002.
3. Herbert Schildt, “The Complete Reference: C#”, Tata McGraw Hill,2004.
4. Andrew Troelsen, “C# and the .NET Platform”, AI Press,2003.

COURSE-XII

SENSOR NETWORKS

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Understand sensor networks as interconnected systems of sensors that collect and transmit data.
- Explore different types of sensors used in sensor networks, such as temperature, humidity, and motion sensors.
- Learn about sensor network architectures, including node communication protocols and data aggregation techniques.
- Study applications of sensor networks in various fields, such as environmental monitoring, smart cities, healthcare, and agriculture.
- Recognize challenges in sensor networks, such as energy efficiency, data security, scalability, and sensor placement optimization.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the fundamental concepts and architecture of sensor networks, including sensor nodes, communication protocols, and network topologies.

CO2: Design and deploy sensor networks for various applications, including environmental monitoring, healthcare, and smart cities.

CO3: Implement and optimize communication protocols suitable for sensor networks, such as Zigbee, Bluetooth Low Energy (BLE), and LoRaWAN.

CO4: Develop energy-efficient algorithms and protocols for data aggregation, routing, and localization in sensor networks.

CO5: Evaluate the performance and scalability of sensor networks using simulation tools and real-world deployments.

Units	Course Content	Teaching Hours
Unit 1	**Basics of Wireless Sensors and Lower Layer Issues:** Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Upper Layer Issues of WSN**: Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, Sensor Networks and mobile robots.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Routing Protocols:** Issues in designing a routing protocol, classification of routing protocols, table-driven, on-demand, hybrid, flooding, hierarchical, and power aware routing protocols.	**10 Hours L(8):T(0):P(2)**
Unit 4	**QoS and Energy Management:** Issues and Challenges in providing QoS, classifications, MAC, network layer solutions, QoS frameworks, need for energy management, classification, battery, transmission power, and system power management schemes.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN–978-1-55860-914-3(Morgan Kauffman).
2. Feng Zhao and Leonides Guibas,"Wireless sensor networks ", Elsevier publication-2004.
3. Jochen Schiller, "Mobile Communications", Pearson Education, 2nd Edition, 2003.

Reference Books:

1. William Stallings, "Wireless Communications and Networks ", Pearson Education – 2004.
2. C. Siva Ram Murthy, and B. S. Manoj, "Ad-Hoc Wireless networks", Pearson Education - 2008.

COURSE-XIII

DATA MINING AND DATA WAREHOUSING

CREDIT (L: T: P)	3:0:1

Objectives:

- To understand the principles of Data warehousing and Data Mining.
- To be familiar with the Data warehouse architecture and its Implementation.
- To know the Architecture of a Data Mining system.
- To understand the various Data Preprocessing Methods.
- To perform classification and prediction of data.

Course outcomes:

At the end of the course the students will be able to

CO1: Recognize and define data mining problems, and implement data warehousing solutions.
CO2: Develop and formulate association rules based on given data patterns.
CO3: Identify, extract, and evaluate interesting patterns across various datasets.
CO4: Design and implement data Preprocessing techniques to prepare data for mining.
CO5: Utilize data mining algorithms to solve real-world problems and analyze the results effectively.

Units	Course Content	Teaching Hours
Unit 1	**Data Warehousing &modelling:** Basic Concepts, Data Warehousing, A multitier Architecture, Data warehouse models, Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube, A multidimensional data model, Stars, Snowflakes and Fact constellations, Schemas for multidimensional Data models, Dimensions, The role of concept Hierarchies, Measures, Their Categorization and computation, **OLAP:** Typical OLAP Operations. Efficient Data Cube computation: An overview, Indexing OLAP Data, Bitmap index and join index, Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Data mining and Association Analysis:** Introduction, Challenges, Data Mining Tasks, Types of Data, Data Quality, Data Preprocessing, Cleaning, Integration, Reduction–PCA, Data Transformation and Discretization, Measures of Similarity and Dissimilarity. **Association Analysis**: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FPGrowth Algorithm, Evaluation of Association Patterns.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Classification:** Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbour Classifiers, Bayesian Classifiers.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Clustering Analysis:** Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Jiawei Han, Micheline Kamber and Jian Pei "Data Mining Concepts and Techniques", Third Edition, Elsevier, 2011.

Reference Books:

1. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.
2. K. P. Soman, Shyam Diwakar and V. Ajay, "Insight into data mining Theory and Practice", Prentice Hall of India, 2006.
3. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Pearson, First impression,2014.
4. Jiawei Han, Micheline Kamber, Jian Pei: Data Mining -Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.

COURSE-XIV

CLOUD COMPUTING

CREDIT (L: T: P) **3:0:1**

Objectives:

- Understand cloud computing as a technology that provides on-demand access to computing resources over the internet.
- Explore different cloud computing service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- Learn about cloud deployment models, including public, private, hybrid, and multicloud environments.
- Study cloud computing benefits such as scalability, flexibility, cost-effectiveness, and accessibility.
- Recognize challenges in cloud computing, such as data security, compliance, vendor lock-in, and performance issues.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the fundamental concepts and principles of cloud computing, including service models (IaaS, PaaS, SaaS) and deployment models (public, private, hybrid).

CO2: Design and deploy applications on cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP).

CO3: Implement cloud-based storage solutions, databases, and data management techniques.

CO4: Utilize containerization technologies like Docker and orchestration tools like Kubernetes for deploying and managing cloud applications.

CO5: Evaluate cloud computing architectures, security issues, and cost optimization strategies for different use cases and industries.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Cloud Computing**: Definition and Characteristics of Cloud Computing, History and Evolution of Cloud Computing, Cloud Computing Architecture, Benefits and Challenges of Cloud Computing.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Cloud Service Models**: Infrastructure as a Service (IaaS), Key Providers: AWS EC2, Google Compute Engine, Azure VMs, Platform as a Service (PaaS), Key Providers: AWS Elastic Beanstalk, Google App Engine, Azure App Services Software as a Service (SaaS), Examples: Google Workspace, Microsoft Office 365.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Cloud Deployment Models:** Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Concept of Virtualization, Types of Virtualizations: Server, Network, Storage, Hypervisors: VMware, Hyper-V, KVM.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Cloud Storage and Databases**: Storage Solutions, S3, Azure Blob Storage, Google Cloud Storage, Database Services AS AWS RDS, Azure SQL Database, Google Cloud SQL, NoSQL Databases as DynamoDB, Azure Cosmos DB, Google Cloud Fire store. **Cloud Networking:** Networking Basics in the Cloud, Virtual Private Cloud (VPC), Load Balancing and Auto Scaling, Content Delivery Networks (CDN). **Introduction to DevOps in the Cloud:** DevOps Principles and Practices, CI/CD Pipelines, Infrastructure as Code (IaC).	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl.

Reference Books:

1. AWS Documentation, Azure Documentation, Google Cloud Documentation, **Tools:** AWS Free Tier, Azure Free Account, Google Cloud Free Tier.

COURSE-XV

MOBILE COMMUNICATIONS

CREDIT (L: T: P)	3:0:1

Objectives:

- Understand mobile communications as the technology enabling wireless transmission of voice, data, and multimedia between mobile devices.
- Explore mobile communication standards such as 4G LTE, 5G, and Wi-Fi, and their capabilities in terms of speed, latency, and coverage.
- Learn about mobile network architectures including cellular networks (e.g., GSM, CDMA) and their evolution towards virtualized and software-defined networks.
- Study mobile communication technologies enabling services like mobile broadband, VoIP, messaging, and mobile applications.
- Recognize challenges in mobile communications, such as spectrum allocation, network congestion, security threats, and regulatory issues.

Course outcomes:

At the end of the course, students will be able to:

CO1: Understand the fundamental concepts and principles of mobile communications, including cellular network architecture and mobile technologies (2G, 3G, 4G, 5G).

CO2: Analyze and compare different mobile communication standards and protocols.

CO3: Design and implement mobile applications for various platforms (iOS, Android) using appropriate development frameworks and tools.

CO4: Evaluate and optimize mobile network performance, including issues related to bandwidth, latency, and reliability.

CO5: Explore emerging trends and technologies in mobile communications, such as Internet of Things (IoT) integration and mobile edge computing.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Mobile Communication**: History and evolution of mobile communication, Overview of mobile communication systems, Key concepts and terminology, Wireless Communication Fundamentals, Radio wave propagation, Modulation and demodulation techniques, Multiple access techniques: FDMA, TDMA, CDMA, and OFDMA	**10 Hours L(8):T(0):P(2)**
Unit 2	**Cellular Concepts:** Cellular architecture and design, Frequency reuse and cell splitting, Handoff strategies, GSM (Global System for Mobile Communications), GSM network architecture, GSM protocols and services, Security in GSM	**10 Hours L(8):T(0):P(2)**
Unit 3	**CDMA (Code Division Multiple Access):** CDMA principles and architecture, CDMA network operations, Comparison with GSM, 3G Mobile Communication Systems, Introduction to 3G networks, UMTS and WCDMA, 3G services and applications	**10 Hours L(8):T(0):P(2)**
Unit 4	**4G Mobile Communication Systems**: Overview of 4G networks, LTE architecture and protocols, Advanced features and services in 4G, 5G Mobile Communication Systems, Introduction to 5G technology, 5G architecture and key technologies, Use cases and applications of 5G **Mobile Communication Protocols**: Signaling protocols (SS7, SIP, etc.), Data protocols (GPRS, EDGE, HSPA), Mobile IP and mobility management **Wireless Networking and Standards:** IEEE 802.11 (Wi-Fi) standards, Bluetooth and other short-range wireless technologies, IoT and M2M communication.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. "Wireless Communications: Principles and Practice" by Theodore S. Rappaport.

Reference Books:

1. Research papers and articles provided throughout the course, **Software tools**: MATLAB, NS-3, or other relevant simulation tools

COURSE-XVI

PROBABILITY AND STATSTICS

CREDIT (L: T: P) **3:1:0**

Objectives:

- Understand probability as the mathematical framework for quantifying uncertainty and randomness.
- Explore basic concepts in probability theory such as probability distributions, random variables, and expected values.
- Learn about statistical methods for data analysis including descriptive statistics, hypothesis testing, and regression analysis.
- Study applications of probability and statistics in various fields such as science, engineering, economics, and social sciences.
- Recognize the importance of probability and statistics in making informed decisions, modeling real-world phenomena, and drawing meaningful conclusions from data.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand and apply basic concepts in probability theory, including probability distributions, random variables, and expected values.

CO2: Apply statistical methods for data analysis, including descriptive statistics, hypothesis testing, and confidence intervals.

CO3: Analyze and interpret data using appropriate statistical techniques, such as regression analysis and ANOVA.

CO4: Utilize statistical software (e.g., R, Python with libraries like NumPy and Pandas) to perform data analysis and visualization.

CO5: Apply probability and statistics to real-world problems in various domains, including business, engineering, and social sciences.

Units	Course Content	Teaching Hours
Unit 1	**Data and Representation**: Introduction to Statistics, Collection of data, classification and tabulation of data, Types of data, Primary data, Secondary data, Presentation of data Diagrammatic and Graphical Representation: Histogram, frequency curve, frequency polygon, Ogive curves, stem and leaf chart. **Measures of Central Tendency:** Mean (A.M.) Definition, Mode, Median, Partition Values: Quartiles, Deciles and Percentiles, Box Plot, Percentile ranks. Means of transformed data, Geometric Mean (G.M.) Definition, Harmonic Mean (H.M.), Weighted A.M., G.M. and H.M.	**10 Hours L(8):T(2):P(0)**
Unit 2	**Dispersion Arithmetic:** Range, Mean deviation Mean square deviation, Variance and Standard Deviation, Combined variance (derivation for 2 groups), Combined standard deviation. **Correlation and Regression**: Bivariate normal distribution, types, importance, methods of measuring correlation-scatter diagram, Karl Pearson's Coefficient of Correlation and Spearman's rank Correlation. Regression lines, Difference between regression and correlation, uses of Regression	**10 Hours L(8):T(2):P(0)**
Unit 3	**Sampling theory and tests of significance:** Methods of sampling (Description only): Simple random sampling with and without replacement (SRSWR and SRWOR) stratified random sampling, systematic sampling. Tests of significance – z, t, chi-square and F.	**10 Hours L(8):T(2):P(0)**
Unit 4	**Distributions:** Random variable, discrete distribution, Continuous distribution, Joint and Conditional distribution, Sampling distributions and applications, Distributions of functions of random variables, Estimation and inference, Multivariate distribution, Compound distribution.	**10 Hours L(8):T(2):P(0)**

Textbooks:

1. Probability and Statistics with applications to Computer Science by K. S. Trivedi

COURSE-XVII

E-commerce and E-Governance

CREDIT (L: T: P)	3:0:1

Objectives:

- ➢ Understand the fundamental principles, models, and benefits of both e-commerce (B2B, B2C, C2C) and e-governance (G2C, G2B, G2G).
- ➢ Explore the technologies and platforms used in e-commerce and e-governance, including web technologies, payment gateways, and technological infrastructure.
- ➢ Learn strategies for online marketing, customer relationship management, supply chain management in e-commerce, and effective public service delivery in e-governance.
- ➢ Study the legal, ethical, and implementation challenges, such as data privacy, consumer rights, digital divide, and data security in both e-commerce and e-governance.
- ➢ Examine real-world applications, case studies, and successful initiatives in e-commerce across various industries and e-governance projects worldwide.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the principles and mathematics behind elliptic curve cryptography.

CO2: Implement the Elliptic Curve ElGamal encryption and decryption algorithms.

CO3: Analyze the security properties of Elliptic Curve ElGamal, including its resistance to attacks such as brute force and elliptic curve discrete logarithm problem.

CO4: Apply Elliptic Curve ElGamal in practical scenarios for secure communication and digital signatures.

CO5: Evaluate the performance and efficiency of Elliptic Curve ElGamal compared to other cryptographic schemes.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to E-commerce:** Fundamentals of E-commerce, E-commerce Business Models, E-commerce Infrastructure, E-commerce Security and Payment Systems.	**10 Hours L(8):T(0):P(2)**
Unit 2	**E-commerce Strategies and Applications:** E-commerce Marketing and Advertising, Supply Chain Management in E-commerce, Emerging Trends in E-commerce, Case Studies and Practical Applications.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Introduction to E-Governance:** Fundamentals of E-Governance, E-Governance Models and Frameworks, Technology and Infrastructure for E-Governance, E-Governance Services and Applications.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Challenges and Future Trends in E-Governance:** Challenges in E-Governance, E-Governance in Developing Countries, Emerging Technologies in E-Governance, Future Directions and Innovations.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. "E-Commerce 2020: Business, Technology, and Society" by Kenneth C. Laudon and Carol Guercio Traver, Pearson.
2. "Electronic Commerce 2018: A Managerial and Social Networks Perspective" by Efraim Turban, Jon Outland, David King, Jae Lee, Ting-Peng Liang, and Deborrah C. Turban, Springer.
3. "E-Business and E-Commerce Management: Strategy, Implementation and Practice" by Dave Chaffey, Pearson.

Reference Books:

1. "E-Government: Information, Technology, and Transformation" by Hans J. Scholl, Routledge.
2. "E-Governance: Managing or Governing?" by Jeremy Millard, Routledge.
3. "Public Information Technology and E-Governance: Managing the Virtual State" by G. David Garson, Jones & Bartlett Learning.
4. "Electronic Governance and Cross-Boundary Collaboration: Innovations and Advancing Tools" by Sharon S. Dawes, IGI Global.
5. "E-Government: Principles and Practices" by Ake Grönlund and Thomas A. Horan, Springer

COURSE-XVIII

INFORMATION RETRIVAL

CREDIT (L: T: P)	3:0:1

Objectives:

- Understand Information Retrieval (IR) as the process of accessing and retrieving relevant information from large collections of data or documents.
- Explore diverse applications of IR in web search engines, digital libraries, and enterprise information systems.
- Learn about IR techniques such as indexing for organizing data and query processing to match user queries with relevant information.
- Study methods in IR including keyword-based searching, natural language processing (NLP), and machine learning algorithms to enhance search accuracy.
- Recognize challenges in IR, such as handling large datasets, ensuring query precision, managing ambiguous queries, and maintaining relevance in dynamic information environments.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of information retrieval (IR).

CO2: Analyze and apply different IR models, such as Boolean, vector space, and probabilistic models.

CO3: Implement and evaluate basic IR techniques, including indexing, querying, and relevance ranking.

CO4: Utilize IR evaluation metrics and methodologies to assess the effectiveness of retrieval systems.

CO5: Apply advanced IR techniques, such as web search, query expansion, and clustering, to real-world datasets.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** architecture of retrieval system, retrieval vs search, inductive and deductive way of retrieval, introduction to multimedia retrieval systems, validity measures- accuracy, precision, recall, f-measure, r-norm. sequencing with grouping, correct index.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Review of the Relational Model**: A Historical Progression, Information Retrieval as a Relational Application, Semi-Structured Search using a Relational Schema, Data Normalization.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Data Clustering:** Data, Features, Feature Space, Data Reduction, Proximity Indices and Similarity/Dissimilarity measures, Fuzzy Measures, Symbolic Measures, Clustering Strategies-Agglomerative Clustering, Divisive Clustering, Partitional Clustering, Cluster Validity, Applications of Data Clustering. Reduction through transformations.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Text retrieval system:** building up a corpus of text documents-preprocessing, feature extraction, different similarity and dissimilarity measures, searching a document based on query, approximations, dimensionality reduction through term selection, term elimination and combination of both, Hashing and indexing for quick retrieval, insertion and deletion of document from the corpus.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. David A. Grossman, Ophir Frieder- Information Retrieval: Algorithms and Heuristics, Second Edition, The Information Retrieval Series, Vol. 15, Springer-2004.

Reference Books:

1. Anil K Jain, R. C. Dubes: Algorithms for Clustering Data.

COURSE-XIX

JAVA PROGRAMMING

CREDIT (L: T: P)	2:0:2

Objectives:

- Understand the basics of Java programming, including syntax, data types, variables, and control structures.
- Explore Object-Oriented Programming (OOP) principles in Java, including classes, objects, inheritance, polymorphism, and encapsulation.
- Learn about error handling and debugging techniques to develop robust and error-free Java applications.
- Study the creation of graphical user interfaces (GUIs) using Java libraries such as Swing and JavaFX.
- Recognize the importance of Java libraries and frameworks, such as the Java Standard Library, Spring, and Hibernate, in enhancing application development.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts of Java programming language, including syntax, data types, control structures, and object-oriented programming principles.

CO2: Develop console applications, graphical user interfaces (GUIs), and web applications using Java.

CO3: Implement and utilize Java features such as collections framework, generics, and lambda expressions.

CO4: Design and develop multithreaded applications using Java concurrency utilities.

CO5: Apply best practices for software development, including debugging, testing, and documenting Java programs.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Java:** Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java, Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Type Conversion and Casting, Automatic Type Promotion in Expressions, A Few Words About Strings	**10 Hours L(5):T(0):P(5)**
Unit 2	**Objects and Classes:** Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character **Operators:** Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The **?** Operator, Operator Precedence, Using Parentheses. **Control Statements:** Java's Selection Statements, Iteration Statements, Jump Statements.	**10 Hours L(5):T(0):P(5)**
Unit 3	**Event and GUI programming**: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle.	**10 Hours L(5):T(0):P(5)**
Unit 4	**Packages and Interfaces:** Packages, Access Protection, Importing Packages, Interfaces. Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Chained Exceptions, Using Exceptions **I/O Programming**: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files. Multithreading in Java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try-catch-finally, Collections in java, Introduction to JavaBeans.	**10 Hours L(5):T(0):P(5)**

Textbooks:

1. Y. Daniel Liang, Introduction to Java Programming (Comprehensive Version), Seventh Edition, Pearson.
2. Sachin Malhotra,Saurabh Chaudhary, Programming in Java, Oxford University Press.
3. Doug Lowe, Joel Murach, Andrea Steelman, Murach's Beginning Java 2, SPD.

Reference Books:

1. Horstmann, Cornell, Core Java Volume-I Fundamentals, Eight Edition, Pearson Education.
2. Herbert Schild, The Complete Reference, Java 2 (Fourth Edition), TMH.
3. D. S. Malik, Java Programming, Cengage Learning.

COURSE-XX

COMPILER CONSTRUCTION

CREDIT (L: T: P)	2:1:1

Objectives:

- Understand the fundamental concepts of compiler construction, including lexical analysis, syntax analysis, semantic analysis, optimization, and code generation.
- Explore the structure and components of a compiler, such as the front-end (lexer and parser).
- Learn about various parsing techniques, including top-down and bottom-up parsing.
- Study optimization techniques used in compilers to improve the efficiency and performance of the generated code.
- Recognize the challenges and solutions in compiler design, such as error detection and recovery, symbol table management, and runtime environment handling.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and phases of compiler construction, including lexical analysis, syntax analysis, semantic analysis, optimization, and code generation.

CO2: Design and implement lexical analysers using tools like Lex to perform tokenization.

CO3: Develop syntax analyzers (parsers) using parsing techniques such as LL, LR, and LALR.

CO4: Apply semantic analysis techniques to perform type checking, scope resolution, and intermediate code generation.

CO5: Implement optimization techniques to improve the performance of generated code, including loop optimization, constant folding, and dead code elimination.

Units	Course Content	Teaching Hours
Unit 1	**Introduction**: language processing system, the phases of a compiler- Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation, Code Optimization, Code Generation, Symbol-Table Management, The Grouping of Phases into Passes, Compiler-Construction Tools, The Evolution of Programming Languages, The Science of Building a Compiler, Modeling in Compiler Design.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Lexical analysis:** tokens, patterns, and lexemes, lexical errors, recognition of tokens, transition diagrams, nondeterministic finite automata, transition tables, deterministic finite automata, conversion of an NFA to a DFA, construction of an NFA from a regular expression, computing nullable, firstpos, and lastpos, computing follow-ups, converting a regular expression directly to a DFA.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Syntax analysis**: context-free grammars, parse trees and derivations, ambiguity, eliminating ambiguity, elimination of left recursion, top-down parsing, recursive-descent parsing, first and follow, LL(l) grammars, predictive parsing, bottom-up parsing, reductions, handle pruning, shift-reduce parsing, LR (0), constructing SLR-parsing tables, canonical LR(l) parsing, LALR parsing.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Syntax-directed definition and translation:** syntax - directed definitions, evaluating an SDD, dependency graphs, S-attributed definitions, L-attributed definitions. Intermediate-code generation, Three-address code representation, Code generation and optimization: Issues in the design of a Target code generator, input to the code generator, the target program.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Monica S Lam, Alfred W Aho, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, Pearson, 2008, 2nd edition.
2. Andrew W Apple, Modern Complier Implementation in C, Cambridge University Press, 2004

Reference Books:

1. Kenneth C Louden, Complier Construction Principles, Thomson Education.

COURSE-XXI

COMPUTER GRAPHICS

CREDIT (L: T: P)	**2:1:1**

Objectives:

- Understand the fundamental concepts and principles of computer graphics, including graphics systems, display devices, and graphics software.
- Explore the mathematics of computer graphics, such as transformations, projections, and viewing, to manipulate and view 2D and 3D objects.
- Learn about raster graphics techniques, including algorithms for drawing lines, circles, and polygons, as well as filling and clipping operations.
- Study the methods for modeling and rendering 3D objects, including shading, lighting, texture mapping, and ray tracing.
- Recognize the applications and challenges in computer graphics, such as animation, simulation, virtual reality, and real-time rendering.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of computer graphics, including graphics hardware, software, and applications.

CO2: Implement algorithms for basic graphics primitives, including line drawing, circle drawing, and polygon filling.

CO3: Apply geometric transformations such as translation, rotation, scaling, and shearing to graphical objects.

CO4: Develop and utilize algorithms for 3D graphics, including viewing transformations, projection, and hidden surface removal.

CO5: Create realistic images using lighting, shading, and texture mapping techniques.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Computer Graphics:** Fundamentals of Computer Graphics, Graphics Hardware and Software, 2D Graphics and Rendering, 3D Graphics and Modeling.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Graphics Algorithms and Techniques:** Geometric Transformations, Rasterization and Clipping, Color Models and Texturing, Hidden Surface Removal and Visibility.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Advanced Rendering Techniques:** Lighting and Shading Models, Global Illumination, GPU Programming and Parallel Rendering, Virtual Reality and Augmented Reality.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Special Topics in Computer Graphics:** Animation Techniques, Advanced Topics in GPU Rendering, Computer Graphics in Games and Simulations, Emerging Trends and Future Directions.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. "Computer Graphics: Principles and Practice" by John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley, Addison-Wesley.
2. "Interactive Computer Graphics: A Top-Down Approach with WebGL" by Edward Angel and Dave Shreiner, Addison-Wesley.
3. "Fundamentals of Computer Graphics" by Peter Shirley, Michael Ashikhmin, Steve Marschner, Erik Reinhard, Kelvin Sung, and Michael Gleicher, A K Peters/CRC Press.

Reference Books:

1. "Introduction to Computer Graphics and the Vulkan API" by Kenwright, P., Independently published.
2. "Real-Time Rendering" by Tomas Akenine-Möller, Eric Haines, and Naty Hoffman, A K Peters/CRC Press.
3. "OpenGL Programming Guide: The Official Guide to Learning OpenGL" by John Kessenich, Graham Sellers, and Dave Shreiner, Addison-Wesley.

COURSE-XXII

OBJECT ORIENTED DESIGN PATTERNS

CREDIT (L: T: P)	3:0:1

Objectives:

- Understand the fundamental principles of object-oriented design, including encapsulation, inheritance, and polymorphism.
- Explore various design patterns, such as creational, structural, and behavioural patterns, and their applications in solving common software design problems.
- Learn to implement and apply design patterns like Singleton, Factory, Observer, Strategy, and Decorator in real-world scenarios.
- Study the benefits of using design patterns to enhance code reusability, maintainability, and scalability.
- Recognize the challenges and best practices in selecting and integrating design patterns into software projects.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the principles and benefits of object-oriented design patterns.

CO2: Identify and describe common design patterns such as creational, structural, and behavioral patterns.

CO3: Apply design patterns to solve software design problems effectively and efficiently.

CO4: Implement design patterns using object-oriented programming languages like Java or C++.

CO5: Evaluate and compare different design patterns based on their suitability and trade-offs in various contexts.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Object-Oriented Design and Principles:** Fundamentals of Object-Oriented Design, SOLID Principles, Design Patterns Overview, UML and Design Documentation.	10 Hours L(8):T(0):P(2)
Unit 2	**Creational Design Patterns:** Factory Method Pattern, Abstract Factory Pattern, Singleton Pattern, Builder and Prototype Patterns.	10 Hours L(8):T(0):P(2)
Unit 3	**Structural Design Patterns:** Adapter Pattern, Composite Pattern, Decorator Pattern, Facade and Proxy Patterns.	10 Hours L(8):T(0):P(2)
Unit 4	**Behavioural Design Patterns:** Observer Pattern, Strategy Pattern, Command and Chain of Responsibility Patterns, State and Template Method Patterns.	10 Hours L(8):T(0):P(2)

Textbooks:

1. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Addison-Wesley.
2. "Head First Design Patterns" by Eric Freeman, Bert Bates, Kathy Sierra, and Elisabeth Robson, O'Reilly Media.
3. "Pattern-Oriented Software Architecture Volume 1: A System of Patterns" by Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal, Wiley.

Reference Books:

1. "Design Patterns Explained: A New Perspective on Object-Oriented Design" by Alan Shalloway and James R. Trott, Addison-Wesley.
2. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" by Craig Larman, Prentice Hall.
3. "Refactoring: Improving the Design of Existing Code" by Martin Fowler, Addison-Wesley.

COURSE-XXIV

GRAPH THEORITICAL ALGORITHMS

CREDIT (L: T: P)	**2:1:1**

Objectives:

- ➢ Develop a fundamental understanding of graph theory, including graph representations, types of graphs (e.g., directed, undirected), and basic properties (e.g., connectivity, cycles).
- ➢ Explore classical graph algorithms such as Depth-First Search (DFS) and Breadth-First Search (BFS), and understand their applications in graph traversal and pathfinding.
- ➢ Learn advanced graph algorithms including Dijkstra's algorithm for shortest paths, Prim's algorithm for minimum spanning trees, and Ford-Fulkerson algorithm for maximum flow.
- ➢ Implement and analyze algorithms for graph problems such as topological sorting, Eulerian and Hamiltonian paths/cycles, and graph coloring.
- ➢ Recognize the theoretical foundations and practical implications of graph algorithms in various domains including network analysis, transportation planning, and social network analysis.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and properties of graphs in graph theory.

CO2: Implement and analyze basic graph algorithms, including traversal algorithms (DFS, BFS) and shortest path algorithms (Dijkstra's algorithm, Bellman-Ford algorithm).

CO3: Apply graph algorithms for solving problems such as minimum spanning trees (Prim's and Kruskal's algorithms) and maximum flow problems (Ford-Fulkerson algorithm).

CO4: Develop algorithms for graph coloring, matching, and clique finding.

CO5: Evaluate and compare the performance of graph algorithms using theoretical analysis and practical implementations.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Graph Theory:** Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles, Graph Colouring, and Chromatic Polynomials	**10 Hours L(6):T(2):P(2)**
Unit 2	**Trees:** Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Optimization and Matching:** Dijkstra"s Shortest Path Algorithm, Minimal Spanning Trees – The algorithms of Kruskal and Prim, Transport Networks – Max-flow, Min-cut Theorem, Matching Theory	**10 Hours L(6):T(2):P(2)**
Unit 4	**The Principle of Inclusion and Exclusion:** The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education, 2004.
2. D.S. Chandrasekharaiah: Graph Theory and Combinatorics, Prism, 2020.
3. Chartrand Zhang: Introduction to Graph Theory, TMH, 2006.

Reference Books:

1. Richard A. Brualdi: Introductory Combinatorics, 6th Edition, Pearson Education, 2018.
2. Geir Agnarsson & Raymond Geenlaw: Graph Theory, Pearson Education, 2018.

COURSE-XXV

NEURAL NETWORKS

CREDIT (L: T: P)	**2:1:1**

Objectives:

- Understand the basic principles of artificial neural networks (ANNs), including neuron models, activation functions, and network architectures (e.g., feedforward, recurrent).
- Explore supervised learning algorithms for training ANNs, such as backpropagation and its variants (e.g., stochastic gradient descent), to solve classification and regression tasks.
- Learn about unsupervised learning techniques like clustering and self-organizing maps (SOMs) using neural networks for data exploration and pattern recognition.
- Study advanced topics in neural networks, including convolutional neural networks (CNNs) for image recognition and recurrent neural networks (RNNs) for sequence modeling.
- Recognize the challenges and ethical considerations in deploying neural networks, including issues related to bias, interpretability, and privacy concerns in AI applications.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of artificial neural networks (ANNs).

CO2: Implement feedforward neural networks and train them using algorithms such as gradient descent and backpropagation.

CO3: Explore different activation functions and their impact on neural network performance.

CO4: Design and implement convolutional neural networks (CNNs) for tasks such as image classification and object detection.

CO5: Develop recurrent neural networks (RNNs) and apply them to sequential data tasks like natural language processing and time series prediction.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Neural Networks:** Fundamentals of Neural Networks, Neural Network Architectures, Training Neural Networks, Evaluation and Performance Metrics.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Advanced Neural Network Architectures:** Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs) Autoencoders and Variational Autoencoders.	**10 Hours L(6):T(2):P(2)**
Unit 3	**Advanced Topics in Neural Networks:** Advanced Optimization Techniques, Regularization and Dropout, Deep Reinforcement Learning, Ethical and Social Implications.	**10 Hours L(6):T(2):P(2)**
Unit 4	**Applications and Future Directions:** Natural Language Processing (NLP), Computer Vision Applications, Neural Networks in Healthcare, Emerging Trends and Future Directions	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press.
2. "Neural Networks and Deep Learning: A Textbook" by Charu C. Aggarwal, Springer.
3. "Pattern Recognition and Machine Learning" by Christopher M. Bishop, Springer.

Reference Books:

1. "Neural Networks and Learning Machines" by Simon Haykin, Pearson.
2. "Neural Network Design" by Martin T. Hagan, Howard B. Demuth, and Mark H. Beale, Martin Hagan.
3. "Understanding Machine Learning: From Theory to Algorithms" by Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press.

COURSE-XXVI

FUZZY SETS AND THEORY

CREDIT (L: T: P)	3:0:1

Objectives:

- Develop a fundamental understanding of fuzzy sets theory, including the concept of membership functions and degrees of truth.
- Explore operations on fuzzy sets, such as union, intersection, complement, and the extension principle.
- Learn about fuzzy logic and reasoning, including fuzzy inference systems and methods.
- Study practical applications of fuzzy sets and fuzzy logic in decision making, control systems, pattern recognition, and artificial intelligence.
- Examine advanced topics in fuzzy theory, such as fuzzy clustering, fuzzy neural networks, and fuzzy control systems, focusing on their theoretical foundations and practical implementations.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of fuzzy sets and fuzzy logic.

CO2: Analyze and apply fuzzy set operations, including union, intersection, complement, and extension principle.

CO3: Develop fuzzy inference systems and implement fuzzy logic controllers.

CO4: Apply fuzzy logic to solve real-world problems in decision making, control systems, and pattern recognition.

CO5: Evaluate and compare fuzzy logic systems with traditional crisp logic approaches.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** Historical perspective, utility of fuzzy systems, limitations of fuzzy systems, statistics and random processes, uncertainty in information, fuzzy sets and membership, chance versus fuzziness, sets as points in Hypercube. **Classical Sets and Fuzzy Sets:** classical sets, operations on them, mapping of classical sets to functions, fuzzy sets, fuzzy set operations, properties of fuzzy sets, non-interactive fuzzy sets	**10 Hours L(8):T(0):P(2)**
Unit 2	**Classical Relations and Fuzzy Relations:** Cartesian Product, Crisp Relations, Cardinality of Crisp Relations, Operations on Crisp Relations, and Properties of Crisp Relations, Composition. **Fuzzy Relations;** Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Cartesian Product and Composition, non-interactive Fuzzy Sets.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Membership Functions:** Features of the Membership Function, Standard Forms and Boundaries, Fuzzification, defuzzification to crisp sets, Lambda-Cuts for Fuzzy Sets, Lambda Cuts for Fuzzy Relations, Defuzzification Methods. **Fuzzy Arithmetic and the Extension Principle:** Crisp Functions, Mapping and Relations, Functions of fuzzy Sets, Extension Principle, Fuzzy Transform (Mapping), Practical Considerations. Fuzzy Numbers Interval Analysis in Arithmetic.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Methods of Extension:** Vertex method, DSW Algorithm, Restricted, DSW Algorithm, Comparisons. Fuzzy Vectors. **Fuzzy Rule Based Systems:** Natural Language, Linguistic Hedges, Rule-Based Systems, Canonical Rule Forms, Decomposition of Compound Rules, Likelihood and Truth Qualification, Aggregation of Fuzzy Rules. Graphical Techniques of Inference.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Fuzzy Logic with Engineering Applications Timothy J. Ross Wiley India International edition,2010.

Reference Books:

1. Fuzzy Logic- Intelligence, Control, and information John Yen Reza Langari Pearson Education 1st Edition, 2004
2. Fuzzy Sets and Fuzzy Logic-Theory and Applications George J. KlirBo Yuan Prentice Hall of India 1 st Edition, 2000
3. Fuzzy Mathematical approach to pattern Recognition, S K Pal, and D Dutta Majumder, John Wiley 1986
4. Neuro-fuzzy pattern recognition: methods in soft computing, S K Pal and S Mitra
5. Fuzzy set theory and its applications by H J Zimmermann, Springer Publications

COURSE-XXVII

SOFT COMPUTING TECHNIQUES

CREDIT (L: T: P) **3:0:1**

Objectives:

- Understand fuzzy sets theory, including membership functions and degrees of truth.
- Explore operations on fuzzy sets: union, intersection, complement, and extension principle.
- Learn fuzzy logic and reasoning, including fuzzy inference systems like Mamdani and TSK models.
- Study practical applications of fuzzy logic in decision making, control systems, and pattern recognition.
- Examine advanced topics in fuzzy theory: fuzzy clustering, neural networks, and control systems.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the principles and foundations of soft computing techniques, including fuzzy logic, neural networks, and evolutionary algorithms.

CO2: Apply fuzzy logic to model uncertainty and vagueness in decision-making processes.

CO3: Implement neural networks for pattern recognition and prediction tasks.

CO4: Develop and optimize evolutionary algorithms such as genetic algorithms and particle swarm optimization for solving optimization problems.

CO5: Evaluate and compare different soft computing techniques in terms of their applicability, efficiency, and robustness.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Soft computing:** Neural networks, Symbolic Data, Genetic algorithms, Hybrid systems and its applications. (ANN, FS, GA, SI, ES, Comparing among intelligent systems) **BNN:** introduction, biological inspiration, BNN & ANN, classification, first Generation NN, perceptron, illustrative problems (2nd generation), introduction, BPN, KNN, HNN, BAM, RBF, SVM and illustrative problems.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Symbolic Data:** Symbolic and Classical Data, Categories, Concepts, and Symbolic Objects, Basic Descriptive Statistics: One Variate, Descriptive Statistics: Two or More Variates, Principal Component Analysis, Regression Analysis, **Cluster Analysis:** Dissimilarity and Distance Measures, Clustering Structures, Partitions, Hierarchy – Divisive Clustering, Hierarchy – Divisive Clustering.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Genetic algorithms:** Introduction, Basic operations, Traditional algorithms, Simple GA General genetic algorithms, Operators, Stopping conditions for GA flow. **Swarm Intelligence System:** Introduction, background of SI, Ant colony system Working of ant colony optimization, ant colony for TSP. Unit commitment problem, Particle Swarm Intelligence system, Artificial bee colony system, Cuckoo search system.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Hybrid Systems:** Sequential Hybrid Systems, Auxiliary Hybrid Systems, Embedded Hybrid Systems, Neuro-Fuzzy Hybrid Systems, Neuro-Genetic Hybrid Systems, Fuzzy-Genetic Hybrid Systems.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Principles of Soft computing, Shivanandam, Deepa S. N, Wiley India, 2011/Reprint2014
2. Soft Computing with MATLAB Programming, N. P. Padhy, S.P. Simon, Oxford, 2015.
3. Symbolic Data Analysis: Conceptual Statistics and Data Mining, Editor(s): Edwin Diday, Monique Noirhomme-Fraiture, First published:18 January 2007

Reference Books:

1. Neuro-fuzzy and soft computing, S.R. Jang, C.T. Sun, E. Mizutani, Phi (EEE edition), 2012.
2. Soft Computing, Saroj Kaushik, Sunita Tiwari, McGraw-Hill, 2018.

COURSE-XXVIII

NATURAL LANGUAGE PROCESSING

CREDIT (L: T: P)	2:0:2

Objectives:

- Understand the key concepts and methods used in NLP.
- Apply NLP techniques to pre-process and analyse text data.
- Develop and evaluate NLP models for various applications.
- Gain hands-on experience with popular NLP libraries and tools.
- Understand the ethical considerations and potential biases in NLP systems, such as fairness and privacy concerns.

Course outcomes:

At the end of the course the students will be able to
CO1: Understand the basic terminology and theory behind underlying natural language processing.
CO2: Understand approaches inflectional and derivational morphology and finite state transducers.
CO3: Understand approaches to part of speech tagging, parsing syntax and semantics in NLP.
CO4: Understand basics of large language models and fine tuning LLM.
CO5: Understand the applications of BERT, GPT.

Units	Course Content	Teaching Hours
Unit 1	**Introduction:** Need for processing of natural languages, Language processing levels, Applications of NLP, Ambiguity and uncertainty in language, Regular Expressions, NLP tasks in syntax, semantics and pragmatics, Machine Translation.	**10 Hours L(5):T(0):P(5)**
Unit 2	**Morphological Processing:** Introduction to Corpus, Tokenization, Stemming, Lemmatization Inflectional and Derivational morphology, Morphological parsing, Finite state transducers, N- gram language models, practical illustrations with NLTK. **Python3:** Textual sources, APIs, social media and Web Scraping, practical illustrations with NLTK, Python3, Textual sources, APIs, social media and Web Scraping.	**10 Hours L(5):T(0):P(5)**
Unit 3	**Information Retrieval**: Design features of Information Retrieval Systems, Classical, non-classical, Alternative Models of Information Retrieval, valuation Lexical Resources: World Net-Frame Net- Stemmers. **Part-of-Speech Tagging**: POS Tagger- Research Corpora.	**10 Hours L(5):T(0):P(5)**
Unit 4	**Large Language Models and its Application:** History and evolution of LLMs, Neural Network Architecture Building Blocks for LLMs, LLM models, Transformer Architecture, Training and Fine-tuning LLMs-Data collection, data Preprocessing, and fine-tuning strategies., Transformer variants: BERT, GPT Architecture, XLNet. Exploring GPT-based applications-chatbots, content generation, and sentiment analysis, Advantages and Challenges of LLM, Ethical and Societal Implications	**10 Hours L(5):T(0):P(5)**

Text Books:

1. D. Jura sky and J. H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Pearson Education,2008
2. J. Allen, Natural Language Understanding, Addison Wesley, 2007.
3. Vineet Chaitanya, Rajeev Sangal. Natural Language Processing - A Paninian Perspective by Akshar Bharathi
4. "Speech and Language Processing" by Daniel Jurafsky and James H. Martin
5. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper

References Books:

1. Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing. Pearson Prentice Hall.
2. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
3. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.
4. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
5. Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press.

COURSE-XXIX

ESSENTIAL METHEMATICS

CREDIT (L: T: P)	3:0:1

Objectives:

- Develop a strong foundation in core mathematical concepts, including arithmetic, algebra, geometry, and calculus.
- Enhance problem-solving skills through practical applications of mathematical principles in various contexts.
- Acquire proficiency in mathematical reasoning and logical thinking to analyze and solve complex problems effectively.
- Learn essential statistical methods for data analysis, including descriptive statistics, probability theory, and inferential statistics.
- Explore diverse applications of mathematics in fields such as science, engineering, economics, and computer science.

Course outcomes:

At the end of the course the students will be able to

CO1: Demonstrate proficiency in basic mathematical concepts, including arithmetic, algebra, geometry, and trigonometry.

CO2: Apply mathematical reasoning and problem-solving skills to solve practical problems.

CO3: Interpret and analyze mathematical data and information presented in various forms, including graphs, tables, and equations.

CO4: Utilize mathematical tools and techniques to model and solve real-world problems in different domains.

CO5: Communicate mathematical ideas and solutions effectively, both orally and in writing.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Complex Variables**: Definitions and properties. Modulus and amplitude of a complex number, Argand‟s diagram, De-Moivre‟s theorem (without proof). **Vector Algebra:** Scalar and vectors. Addition and subtraction and multiplication of vectors- Dot and Cross products, problems	**10 Hours L(8):T(0):P(2)**
Unit 2	**Differential Calculus:** Review of successive differentiation-illustrative examples. Maclaurin‟s series expansions-Illustrative examples. Partial Differentiation: Euler‟s theorem-problems on first order derivatives only. Total derivatives-differentiation of composite functions. Jacobians of order two-Problems	**10 Hours L(8):T(0):P(2)**
Unit 3	**Vector Differentiation**: Differentiation of vector functions. Velocity and acceleration of a particle moving on a space curve. Scalar and vector point functions. Gradient, Divergence, Curl-simple problems. Solenoidal and irrotational vector fields-Problems. **Ordinary differential equations (ODE's):** Introduction-solutions of first order and first-degree differential equations: exact, linear differential equations. Equations reducible to exact and Bernoulli‟s equation.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Numerical Methods:** Finite differences. Interpolation/extrapolation using Newton‟s forward and backward difference formulae (Statements only)-problems. Solution of polynomial and transcendental equations – Newton-Raphson and Regula-Falsi methods (only formulae)- Illustrative examples. **Numerical integration**: Simpson‟s one third rule and Weddle‟s rule (without proof) Problems.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. S C Chapra and R P Canale, Numerical Methods for Engineering, 15th Edition, Tata McGraw Hill
2. Erwin Kreyszig, Advanced Engineering Mathematics, Latest edition, Wiley Publications.
3. B.S. Grewal, Higher Engineering Mathematics, Latest edition, Khanna Publishers.

Reference Books:

1. B.V. Ramana, Higher Engineering Mathematics, Latest edition, Tata McGraw Hill.
2. Srimanta Pal & Subodh C. Bhunia: "Engineering Mathematics" Oxford University Press, 3rd Reprint, 2016.

COURSE-XXX

BLOCKCHAIN TECHNOLOGY

CREDIT (L: T: P) **3:0:1**

Objectives:

- Develop a foundational understanding of blockchain technology, covering its principles, components, and decentralized nature.
- Explore various blockchain types (public, private, consortium) and their applications across industries.
- Learn cryptographic techniques integral to blockchain, including hashing, digital signatures, and Merkle trees.
- Study consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS) used for securing blockchain networks.
- Examine practical uses of blockchain beyond cryptocurrencies, including smart contracts, supply chain management, and decentralized finance.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of blockchain technology, including decentralized ledgers, consensus mechanisms, and cryptography.

CO2: Analyze and compare different blockchain platforms and architectures, such as Bitcoin, Ethereum, and Hyperledger.

CO3: Implement and develop smart contracts using blockchain programming languages like Solidity.

CO4: Design and deploy decentralized applications (DApps) on blockchain networks.

CO5: Evaluate security, scalability, and regulatory aspects of blockchain implementations.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Blockchain Technology:** History and Evolution, Origins of Blockchain, Bitcoin and the rise of crypto currencies, Key milestones in blockchain development.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Blockchain Architecture:** Structure of a Blockchain, Blocks, transactions, and chains, Merkle trees, Distributed ledger technology, Consensus Algorithms, Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT).	**10 Hours L(8):T(0):P(2)**
Unit 3	**Cryptography and Security:** Cryptographic Fundamentals, Public-key cryptography, Private keys and addresses, Digital signatures, Security in Blockchain, Double-spending problem, 51% attack, Security of smart contracts.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Blockchain Platforms:** Bitcoin, Bitcoin protocol and network, Bitcoin transactions and scripting, Wallets and mining, **Ethereum:** Ethereum Virtual Machine (EVM), Smart contracts and Solidity, Decentralized applications (DApps), Other Platforms, Hyperledger Fabric, Ripple, EOS, Finance Smart Chain.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Mastering Bitcoin" by Andreas M. Antonopoulos
2. "Mastering Ethereum" by Andreas M. Antonopoulos and Gavin Wood

Reference Books:

1. "Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher.

COURSE-XXXI

DIGITAL SIGNAL PROCESSING

CREDIT (L: T: P)	3:0:1

Objectives:

- ➢ Develop a foundational understanding of digital signal processing (DSP), including sampling, quantization, and digital signal representation.
- ➢ Explore signal analysis techniques such as Fourier transform (DTFT, DFT), and their applications in spectrum analysis and filtering.
- ➢ Learn about digital filter design methods, including FIR and IIR filters, and their implementation in signal processing applications.
- ➢ Study algorithms for digital signal modulation and demodulation, including amplitude modulation (AM), frequency modulation (FM), and digital signal encoding techniques.
- ➢ Examine advanced topics in DSP, such as adaptive signal processing, wavelet transforms, and applications in audio processing, image processing, and telecommunications.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of digital signal processing.

CO2: Analyze and process digital signals using techniques such as filtering, convolution, and Fourier analysis.

CO3: Implement digital signal processing algorithms for tasks such as signal denoising, compression, and feature extraction.

CO4: Design and optimize digital filters using methods like FIR (Finite Impulse Response) and IIR (Infinite Impulse Response).

CO5: Apply DSP techniques to real-world applications in areas such as telecommunications, audio processing, and biomedical signal analysis.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Digital Signal Processing:** Fundamentals of Signals and Systems, Sampling and Discrete-Time Signals, Discrete Fourier Transform (DFT), Introduction to Digital Filters.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Advanced Digital Signal Processing Techniques:** Filter Design and Realization, Multirate Signal Processing, Time-Frequency, Statistical Signal.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Digital Signal Processing Applications:** Audio Signal Processing, Image and Video Processing, Biomedical Signal Processing, Digital Signal Processing in Communications.	**10 Hours L(8):T(0):P(2)**
Unit 4	**Advanced Topics and Emerging Trends:** Adaptive Signal, Sparse Signal, Machine Learning for Signal Processing, Emerging Trends and Future Directions. **Conclusion and Integration:** Review and Synthesis, Hands-on Projects and Case, Future Directions and Career Opportunities.	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Simon Haykin and Barry Van Veen, “Signals and Systems,” 2 nd Edition, 2008, Wiley India. ISBN9971-51- 239-4.
2. Proakis & Manolakis, “Digital Signal Processing - Principles Algorithms & Applications”, 4th Edition, Pearson education, New Delhi, 2007. ISBN: 81-317-1000-9.

Reference Books:

1. Sanjit K Mitra, “Digital Signal Processing, A Computer Based Approach”, 4th Edition, McGraw Hill Education, 2013
2. Oppenheim & Schaffer, “Discrete Time Signal Processing", PHI, 2003.
3. D Ganesh Rao and Vineeth P Gejji, “Digital Signal Processing" Cengage India Private Limited, 2017, ISBN: 9386858231.
4. V. Udayashankara, “Modern Digital Signal Processing”, Third Edition, PHI 2016.

COURSE-XXXII

DISTRIBUTED OPERATING SYSTEM

CREDIT (L: T: P)	**3:0:1**

Objectives:

- Develop a comprehensive understanding of distributed operating systems, covering principles such as transparency, concurrency, and fault tolerance.
- Explore communication mechanisms in distributed systems, including RPC (Remote Procedure Call) and message-passing protocols.
- Learn synchronization techniques for managing concurrent processes in distributed environments, including distributed mutual exclusion and deadlock prevention.
- Study the architecture and principles of distributed file systems, focusing on replication, consistency, and fault tolerance strategies.
- Examine resource management strategies in distributed systems, including load balancing, scheduling algorithms, and distributed transaction processing.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamental concepts and principles of distributed operating systems.

CO2: Analyze and compare different architectures and models of distributed systems, including client-server, peer- to-peer, and cloud computing.

CO3: Implement and manage distributed processes, communication protocols, and synchronization.

CO4: Design and develop distributed file systems and resource management strategies.

CO5: Evaluate performance, fault tolerance, and security issues in distributed operating systems**.**

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Distributed Systems:** Definition and characteristics of distributed systems, Types of distributed systems, Design goals and challenges, Examples of distributed operating systems. **Architectures of Distributed Systems:** Client-server model, Peer-to-peer systems, Middleware and its role in distributed systems, Case studies of distributed operating systems. **Communication in Distributed Systems:** Inter-process communication (IPC), Remote Procedure Calls (RPC), Message passing mechanisms, Socket programming.	**10 Hours L(8):T(0):P(2)**
Unit 2	**Synchronization and Coordination:** Clock synchronization, Logical clocks (Lamport's and vector clocks), Mutual exclusion algorithms, Election algorithms (Bully and Ring algorithms). **Distributed File Systems and Shared Memory:** Distributed file system design, Case studies: NFS, AFS, Google File System, Distributed shared memory, Consistency models and protocols. **Resource Management:** Distributed scheduling, Load balancing algorithms, Distributed deadlocks, Resource allocation and sharing.	**10 Hours L(8):T(0):P(2)**
Unit 3	**Fault Tolerance and Reliability:** Fault models and failure detection, Replication techniques, Checkpointing and rollback recovery, Consensus protocols (Paxos, Raft), **Security in Distributed Systems:** Security threats and challenges, Cryptographic techniques, Authentication and authorization, Secure communication protocols	**10 Hours L(8):T(0):P(2)**
Unit 4	**Performance and Scalability:** Performance metrics and benchmarking, Scalability challenges, Techniques for improving performance, Case studies of scalable distributed systems, **Advanced Topics and Emerging Trends:** Cloud computing and distributed operating systems, Internet of Things (IoT) and edge computing, Block chain and distributed ledger technologies, Future directions in distributed systems	**10 Hours L(8):T(0):P(2)**

Textbooks:

1. Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten Van Steen
2. "Distributed Operating Systems" by Andrew S. Tanenbaum

Reference Books:

1. "Distributed Systems: Concepts and Design" by George Coulouris, Jean Dollimore, and Tim Kindberg.

COURSE-XXXIII

LINUX PROGRAMMING

CREDIT (L: T: P)	2:1:1

Objectives:

- Develop a strong foundation in Linux fundamentals, covering file system navigation, command-line utilities, and basic shell scripting.
- Explore advanced shell scripting techniques to automate system tasks and improve administrative efficiency.
- Learn system programming concepts on Linux, including process management, file I/O operations, and inter-process communication (IPC).
- Acquire skills in network programming with Linux, focusing on socket programming and understanding network protocols.
- Apply programming languages like C/C++, Python, or Shell scripting for developing and deploying applications on the Linux platform.

Course outcomes:

At the end of the course the students will be able to

CO1: Understand the fundamentals of Linux operating system architecture and command-line interface.

CO2: Develop and debug programs using Linux development tools and environments.

CO3: Implement system-level programming tasks in Linux, such as process management, file handling, and inter-process communication (IPC).

CO4: Utilize Linux libraries and APIs for networking, threading, and memory management.

CO5: Write scripts and automate tasks using shell scripting and other scripting languages on Linux.

Units	Course Content	Teaching Hours
Unit 1	**Introduction to Linux Programming:** Overview of Linux operating system, Setting up a Linux development environment, Basic Linux commands and shell scripting, Introduction to GCC and Make file. **Linux System Programming Fundamentals:** System calls and library functions, File I/O operations, Error handling and debugging, Understanding man pages and documentation, File Handling and Filesystem, File descriptors and file control operations, Directory management, Permissions and file attributes, Filesystem hierarchy and navigation.	**10 Hours L(6):T(2):P(2)**
Unit 2	**Process Management:** Process creation and termination, Process states and scheduling, Zombie and orphan processes, Introduction to signals and signal handling. **Inter-Process Communication (IPC):** Pipes and FIFOs, Message queues, Semaphores, Shared memory. **Thread Programming:** Introduction to POSIX threads, Thread creation and management, Synchronization mechanisms (mutexes, condition variables), Thread safety and re-entrancy	**10 Hours L(6):T(2):P(2)**
Unit 3	**Network Programming - Part 1:** Basics of network programming, Sockets API, TCP/IP protocol suite, Creating client-server applications. **Network Programming - Part 2:** Advanced socket programming (non-blocking I/O, select, poll), Multicasting and broadcasting, Secure network programming (SSL/TLS), Debugging and performance tuning. **Advanced Linux Programming Techniques:** Memory management (mmap, malloc, free), Dynamic linking and shared libraries, Handling large files, Introduction to kernel programming	**10 Hours L(6):T(2):P(2)**
Unit 4	**Debugging and Profiling:** Debugging with GDB, Memory leak detection (Valgrind), Performance profiling (gprof, perf), Analyzing and optimizing code. **Development Tools and Best Practices:** Version control with Git, Automated build systems (CMake, Autotools), Code quality and testing (unit tests, integration tests), Documentation and code review practices.	**10 Hours L(6):T(2):P(2)**

Textbooks:

1. Advanced Programming in the UNIX Environment" by W.Richard Stevens, Stephen A. Rago.
2. "Linux System Programming" by Robert Love.

Reference Books:

1. "The Linux Programming Interface" by Michael Kerrisk and Online documentation and man pages.