Post Graduate Department of Computer Sciences, The University of Kashmir, Srinagar-190006

Revised Scheme & Syllabus Of M.Tech CSE Programme

(2024-2026)

Eligibilty for 2- year M.Tech CSE Programme

B.E/B.Tech in Computer Science Engineering/ Information Technology/Computer Engineering with 55% marks in aggregate for general category and 50% marks for reserved category in the qualifying examination from this University or from any other university as equivalent thereto by this University.

To be effective from year-2024

General Guidelines for formulating the syllabus of M.Tech Programme

**This template is to be followed by every faculty member.*

1. Every Post-graduate program has its own objectives and educational outcomes. These objectives and outcomes are furnished by considering various aspects and impacts of the curriculum. These **Program Outcomes (POs)** are categorically mentioned at the beginning of the curriculum (ref: NBA Manual). There should always be a rationale and a goal behind the inclusion of a course in the curriculum. Course Outcomes though highly rely on the contents of the course; many-a-times are generic and bundled. The **Course Objectives**, **Course Outcomes** and **CO-PO mappings matrix** justifies the motives, accomplishment and prospect behind learning the course. The Course Objectives, Course Outcomes and CO-PO Mapping Matrix are provided for reference and these are indicative only. The course instructor may modify them as per his or her perspective.

2. **CO and PO Mapping Matrix** (Course Outcomes and Program Outcomes)- The expected attainment mapping matrix at end of course contents, indicates the correlation levels of 3, 2, 1 and '-'. The notation of 3, 2 and 1 denotes substantially (high), moderately (medium) and slightly (low). The mark '-'indicates that there is no correlation between the respective CO and PO.

3. For each unit contents, the desired content attainment mapping is indicated with Course Outcome(s).

4. Lab Instructors may design suitable set of assignments for respective course at their level. Beyond curriculum assignments and mini-project may be included as a part of lab work. The Inclusion of few optional assignments that are intricate and/or beyond the scope of curriculum will surely be the value addition for the students and it will satisfy the intellectuals within the group of the learners and will add to the perspective of the learners.

5. Lab Manual- Program codes with sample output of all performed assignments are to be submitted as softcopy. Use of DVD or similar media containing student's programs maintained by Lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Lab. Submission of journal in the form of softcopy is desirable and appreciated

6. Tutorial- Tutorials can never be an individual course but an additional aid to the learners. Tutorials help the learners to inculcate the contents of the course with focused efforts on small group of the learners. Tutorial conduction should concentrate more on simplifying the intricacies converging to clear understanding and application.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

Program Educational Objectives for M.Tech in Computer Science and Engineering

PEO1: To equip students with advanced knowledge and skills to secure employment in industry, academia, and research institutions, and to excel in postgraduate and doctoral studies in the domain of Computer Science and Engineering, while fostering an entrepreneurial mindset and innovation.

PEO2: To prepare students to critically analyze and solve complex computing problems by employing advanced methodologies, innovative algorithms, and cutting-edge technologies in

PEO3: To train students to effectively plan, manage, and lead projects by integrating technical and managerial skills, enabling them to function proficiently as individuals, team members, and leaders in multidisciplinary environments, and to drive innovation and entrepreneurship.

PEO4: To foster an academic environment that encourages excellence and innovation, and to instill in students the importance of lifelong learning, continuous professional development, and the pursuit of entrepreneurial opportunities in a rapidly evolving global technological landscape.

PROGRAM OUTCOMES (POs)

Program Outcomes for M.Tech in Computer Science and Engineering

1. Computing Knowledge: Apply the advanced knowledge of computer science, mathematics, and engineering fundamentals to solve complex problems in software engineering and related fields.

2. Problem Analysis: Identify, formulate, and analyze complex computing problems by reviewing research literature and using first principles of computer science, algorithms, and software engineering sciences to reach substantiated conclusions.

3. Design/Development of Solutions: Design innovative computer based solutions for complex computing problems, considering public ethics, health and safety, and cultural, societal, and environmental factors while meeting specified requirements.

4. Conduct Investigations of Complex Problems: Utilize advanced research-based knowledge and methods, including experimental design, data analysis, and information synthesis, to investigate and provide valid conclusions for complex computing problems.

5. Modern Tool Usage: Develop and apply advanced techniques, tools, and resources, including predictive modeling and modern software engineering and development tools, to complex computing activities with an understanding of their limitations.

6. The Software Engineer and Society: Assess societal, ethical, legal, and cultural issues using contextual knowledge and apply this understanding to professional software development practices, recognizing the associated responsibilities.

7. Environment and Sustainability: Evaluate the impact of professional software engineering solutions on society and the environment, demonstrating a commitment to sustainable development practices.

8. Ethics: Adhere to and promote ethical principles and professional responsibilities in software engineering practices, with a strong emphasis on maintaining privacy and data security.

9. Individual and Team Work: Work effectively as an individual, as well as a member or leader of diverse and multidisciplinary teams, demonstrating strong collaboration and leadership skills.

10. Communication: Communicate complex technical concepts and solutions effectively with peers, professionals, and the broader community, through well-organized reports, design documentation, presentations, and clear instructions.

11. Project Management and Finance : Apply advanced knowledge of software engineering and management principles to plan, manage, and execute projects efficiently, considering financial constraints and resource management in multidisciplinary environments.

To be effective from year-2024

12. Life-long Learning: Recognize and engage in continuous learning and professional development to keep up with the evolving technological landscape and advancements in computer science and engineering.

These outcomes ensure that graduates are equipped with the necessary skills and knowledge to excel in their professional careers and contribute effectively to the advancement of the computer science and engineering field.

Program Specific Outcomes:

1. Advanced Software Engineering Proficiency: Graduates will exhibit creativity, imagination, and advanced proficiency as software engineers, enabling them to excel in industry roles, government positions, and related services, applying sophisticated computing solutions and methodologies.

2. Research and Academic Excellence: Graduates will possess the capability to advance in academic and research pursuits within the field of computer science and engineering, contributing to innovative research and development in computing technologies and related areas.

3. Leadership in Innovation and Entrepreneurship: Graduates will demonstrate leadership in driving innovation and entrepreneurship, upholding high standards of professional and ethical conduct, and making meaningful contributions to societal advancement through new technologies and entrepreneurial initiatives.

These outcomes ensure that graduates are equipped with the necessary skills and knowledge to excel in their professional careers and contribute effectively to the advancement of the computer science and engineering field.

Semester I

To be effective from year-2024

COURSE TITLE: Mathematical Foundation of Computer Science									
Course Code:				**CSE-24101C**			**Examination Scheme**		
Total number of Lecture Hours: 56							**External**	**80**	
							Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**0**	**Tutorial (T):**	**0**		**Total Credits**	**4**	

Course Objectives

- Core Knowledge: Apply computer science and math to solve complex problems.
- Problem Analysis: Analyze issues using algorithms and computational methods.
- Solution Design: Create ethical and efficient computing solutions.
- Lifelong Learning: Continuously update computer science skills

Course Content	TEACHING HOURS
UNIT 1	**14 Hrs**
Core Probability Concepts: Probability mass functions (PMF), probability density functions (PDF), and cumulative distribution functions (CDF). Distribution Families: Key parametric families (e.g., normal, binomial, and Poisson distributions) .Moments and Expectations: Expected value, variance, and conditional expectation. Limit Theorems and Probabilistic Inequalities: Applications of the Central Limit Theorem (univariate and multivariate).	
UNIT 2	**14 Hrs**
Sampling Theory: Understanding random samples and the sampling distributions of estimators, Estimation Techniques : Methods of Moments and Maximum Likelihood Estimation (MLE), Applications of Sampling	
UNIT 3	**14 Hrs**
Fundamentals of Statistical Inference: Concepts for drawing conclusions from data, including hypothesis testing. Multivariate Models: Basics of regression and classification for structured data. Principal Component Analysis (PCA): Introduction to PCA for dimensionality reduction and its importance.	
UNIT 4	**14 Hrs**
Model Assessment Techniques: Techniques to evaluate model performance, such as cross validation and metrics for accuracy. Over-fitting and Regularization: Understanding the issue of overfitting and strategies to mitigate it. Introduction to Markov Chains.	

Textbooks

- Probability and Statistics with Reliability, Queuing, and Computer Science Applications by Kishor S. Trivedi, 2nd Edition, PHI Learning, 2021.
- M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2nd Edition, 2017.
- Fundamentals of Probability and Statistics for Engineers by T.T. Soong, Wiley India, latest edition.
- John Vince, Foundation Mathematics for Computer Science, Springer, 2015.

Reference Books

- K. Trivedi, Probability and Statistics with Reliability, Queueing, and Computer Science Applications, Wiley, 2nd Edition, 2016.
- Alan Tucker, Applied Combinatorics, Wiley, 6th Edition, 2012

COURSE OUTCOMES (CO):

CO1: Apply probability concepts in computer science.
CO2: Use sampling and estimation in computational analysis.
CO3: Conduct statistical inference for computing tasks.
CO4: Evaluate and refine computational model performance.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	2	–	1	–	–	–	–	–	–	–	–
2	3	3	2	–	–	–	–	–	–	–	–	–
3	2	3	2	–	–	–	–	–	1	–	–	–
4	2	2	3	2	1	–	–	–	–	–	–	–

COURSE TITLE: Advanced Data Structures

Course Code:				**CSE-24102C**		**Examination Scheme**			
Total number of Lecture Hours: 56						**External**		**80**	
						Internal		**20**	
Lecture (L):	**4**	**Practical (P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**		**6**	

Course Objectives

- To provide students with a comprehensive understanding of basic data structures such as stacks, queues, trees, graphs, and hash tables, including their representations, operations, and applications.
- To enable students to implement and apply advanced data structures like skip lists, AVL trees, red-black trees, and B-trees using both arrays and linked lists.
- To familiarize students with different techniques in algorithm design and optimization, including collision resolution in hashing, graph traversals, and dynamic programming techniques.
- To develop students' ability to apply various data structures and algorithms to real-world problems such as text processing, pattern matching, and compression algorithms.

Course Content	TEACHING HOURS
UNIT 1: Basic Data Structures and Their Implementations	**14 Hrs.**
Basic concepts overview: Stack, Representation of stack in memory, Operations on Stacks, Implementation of Stack using arrays and linked list, Applications of stacks, Queues, Representation of Queue in Memory, Operations on Queue, Implementation of Queue using arrays and linked list, Queues, Implementation using Arrays and Linked list. **Skip Lists:** Need for Randomizing Data Structures and Algorithms, Search and Update Operations on Skip Lists, Probabilistic Analysis of Skip Lists, Deterministic Skip Lists	
UNIT 2: Advanced Tree and Graph Structures	**14 Hrs.**
Trees : Definitions, terminologies and properties, Binary tree representation, traversals and applications, Threaded binary trees, Binary Search Trees, AVL Trees, M-way Search Trees, B-trees, B*-trees. **Graphs:** Terminology, Graph representations, Traversal Techniques, Operations on Graphs, Applications of Graphs Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees	
UNIT 3: Dictionary Structures and Hashing Techniques	**14 Hrs.**
Dictionaries: Definition, Dictionary Abstract Data Type, Implementation of Dictionaries. **Hashing:** Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing.	

UNIT 4 : String Processing and Pattern Matching Algorithms	14 Hrs.
Text Processing: Sting Operations, Brute-Force Pattern Matching, The Boyer- Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Standard Tries, Compressed Tries, Suffix Tries, The Huffman Coding Algorithm, The Longest Common Subsequence Problem (LCS), Applying Dynamic Programming to the LCS Problem.	

Textbooks

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson, 2004.

Reference Books

2. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.

COURSE OUTCOMES (CO):

CO1: Implement and Manipulate Data Structures: Students will be able to implement and perform basic operations (insert, delete, search, etc.) on fundamental data structures like stacks, queues, trees, and graphs using both arrays and linked lists.
CO2: Apply Advanced Algorithms: Students will be able to apply advanced algorithms such as the Boyer-Moore and Knuth-Morris-Pratt pattern matching algorithms, Huffman coding, and the Longest Common Subsequence (LCS) problem, along with understanding their efficiency.
CO3: Analyse and Optimize Hashing Techniques: Students will be able to apply different hashing techniques (e.g., separate chaining, linear probing, double hashing) to resolve collisions and optimize dictionary operations, while understanding their time and space complexities.
CO4: Solve Complex Data Structure Problems: Students will be capable of designing, implementing, and analysing more complex data structures (e.g., skip lists, B-trees, AVL trees) and apply them to solve real-world problems efficiently, leveraging their probabilistic and deterministic properties.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	2	2	1	-	1	1	2	1	2
2	3	3	3	2	2	1	1	1	2	2	3	1
3	3	3	2	2	2	1	1	2	2	2	-	1
4	3	3	3	3	2	-	1	2	2	3	3	3

To be effective from year-2024

Advanced Data Structures Lab

The lab should cover the following topics:

1. Stack Implementation
2. Queue Implementation
3. Skip List Operations
4. Dictionary Implementation
5. Hashing Techniques
6. Binary Tree Operations
7. Binary Search Tree (BST) Operations
8. AVL Tree Operations
9. Graph Representation and Traversal
10. *B-trees and B Trees**
11. Pattern Matching Algorithms
12. Tries and Compressed Tries
13. Huffman Coding Algorithm
14. Longest Common Subsequence (LCS)
15. Graph Algorithms (Advanced)
16. Dynamic Programming (Advanced Algorithms)

Lab Setup Notes:

- Programming Language: These labs can be implemented in programming languages like C, C++, Java, or Python, depending on the course's focus.
- Tools/Software: Students can use IDEs like Eclipse, IntelliJ IDEA, Visual Studio Code, or NetBeans for implementation.

COURSE TITLE: Research Methodology and IPR

Course Code:				**CSE-24103C**		**Examination Scheme**		
Total number of Lecture Hours: 28						**External**	**40**	
						Internal	**10**	
Lecture(L):	**2**	**Practicals(P):**	**-**	**Tutorial(T):**	**-**	**Total Credits**	**2**	

Course Objectives

- **Understand the Foundations of Research:** Develop knowledge about selecting and defining research problems, and approaches to problem-solving in research.
- **Conduct Ethical and Effective Research:** Learn to conduct effective literature reviews, handle data responsibly, and practice ethical research.
- **Comprehend Intellectual Property (IP) Fundamentals:** Understand the basics of patents, copyrights, and trademarks, and their significance in innovation.
- **Learn Patent Processes and Technology Transfer:** Understand patenting processes, international IP laws, and technology transfer for real-world applications.

Course Content	TEACHING HOURS
UNIT 1: Foundations of Research	**14 Hrs**
Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations. Effective literature studies approaches, analysis Plagiarism, Research ethics,	
UNIT 2: Fundamentals of Intellectual Property and Patenting	**14 Hrs**
Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT. Patent Rights: Scope of Patent Rights. Licensing and transfer of technology.	

Textbooks

- Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students'"
- Wayne Goddard and Stuart Melville, "Research Methodology: An Introduction"
- Ranjit Kumar, 2 ndEdition, "Research Methodology: A Step by Step Guide for• beginners"

ReferenceBooks

- Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd,2007
- Mayall, "Industrial Design", McGraw Hill, 1992.
- Niebel, "Product Design", McGraw Hill, 1974.
- Asimov, "Introduction to Design", Prentice Hall, 1962.
-

COURSEOUTCOMES(CO):

- **CO1:** Identify and define appropriate research problems and demonstrate knowledge of ethical research practices.
- **CO2:** Apply methodologies for data collection, analysis, and interpretation within ethical guidelines.
- **CO3:** Explain types of intellectual property and the fundamentals of patents, copyrights, and trademarks.
- **CO4:** Describe the patenting process, international IP frameworks, and principles of technology transfer.

LEVELOFCO-PO MAPPINGTABLE

	POs											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	2	2	2	1	2	-	3	-	2	-	3
2	3	3	3	2	2	2	-	3	1	2	-	3
3	2	1	2	1	2	3	1	3	-	2	-	3
4	2	2	3	2	2	2	1	3	-	3	1	2

Program Elective I

CSE-24104XEXX

Program Elective II

CSE-24105XEXX

Audit Course

CSE-24106

To be effective from year-2024

Semester -II

COURSE TITLE: Advanced Algorithms									
Course Code:				**CSE-24201C**			**Examination Scheme**		
Total number of Lecture Hours: 56							**External**	**80**	
							Internal	**20**	
Lecture (L):	**4**	**Practical (P):**	**-**	**Tutorial (T):**	**-**		**Total Credits**	**4**	

Course Objectives

- Review and understand the design, analysis, and correctness of various classical and advanced algorithms in sorting, graph theory, dynamic programming, and network flows.
- Gain proficiency in applying key algorithmic paradigms such as **greedy algorithms, dynamic programming, divide-and-conquer**, and **linear programming** to solve real-world problems.
- Perform **time and space complexity analysis** and evaluate the efficiency of algorithms using mathematical proofs and amortized analysis.
- Learn advanced techniques such as **Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Chinese Remainder Theorem**, and **polynomial representation** to solve complex computational problems.

Course Content	**TEACHING HOURS**
UNIT 1: Sorting, Graph Algorithms, and Amortized Analysis	**15 Hrs**
Sorting: Review of various sorting algorithms, topological sorting Graph: Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.	
UNIT 2: Matroids, Graph Matching, and Flow Networks	**14 Hrs**
Matroids: Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST. Graph Matching: Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path. Flow-Networks: Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.	
UNIT 3: Dynamic Programming, Number Theory, and Fourier Transforms	**14 Hrs**
Shortest Path in Graphs: Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming. Modulo Representation of integers/polynomials: Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem. Discrete Fourier Transform (DFT): In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm	

UNIT 4: Linear Programming, NP-Completeness, and Advanced Topics	**14 Hrs**
Linear Programming: Geometry of the feasibility region and Simplex algorithm NP-completeness: Examples, proof of NP-hardness and NP-completeness. One or more of the following topics based on time and interest Approximation algorithms, Randomized Algorithms, Interior Point Method, Advanced Number Theoretic Algorithm Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.	

Textbooks

1. "Introduction to Algorithms" by Cormen, Leiserson, Rivest, Stein.

Reference Books

2. "The Design and Analysis of Computer Algorithms" by Aho, Hopcroft, Ullman.
3. "Algorithm Design" by Kleinberg and Tardos.

COURSE OUTCOMES (CO):

CO1: Apply Advanced Algorithms: Solve complex problems in sorting, graph theory, and network flow by applying advanced algorithmic techniques.

CO2: Evaluate and Analyse Algorithm Performance: Analyse and prove the correctness and efficiency of algorithms, providing rigorous time and space complexity analyses.

CO3: Solve Graph Matching and Flow Problems: Implement and apply algorithms for maximum matching in graphs, as well as solve max-flow min-cut problems using the Ford-Fulkerson and Edmond-Karp algorithms.

CO4: Apply Advanced Algorithmic Techniques to Recent Trends: Stay up to date with recent advancements in algorithms, including approximation algorithms, randomized algorithms, and their applications to modern problem-solving scenarios.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	3	3	2	3	1	1	2	2	2	2	2
2	3	3	2	3	3	2	1	2	2	3	2	3
3	3	3	3	3	2	1	1	2	3	2	2	2
4	3	3	2	2	3	2	1	2	2	3	3	3

COURSE TITLE: Soft Computing									
Course Code:				**CSE-24202C**		**Examination Scheme**			
Total number of Lecture Hours: 56						**External**	**80**		
						Internal	**20**		
Lecture (L):	**4**	**Practicals(P):**	**4**	**Tutorial (T):**	**-**	**Total Credits**	**6**		

Course Objectives

- To introduce soft computing concepts and techniques and foster their abilities in designing appropriate technique for a given scenario.
- To implement soft computing based solutions for real-world problems.
- To give students knowledge of non-traditional technologies and fundamentals of artificial neural networks, fuzzy sets, fuzzy logic, genetic algorithms.
- To provide student an hand-on experience on MATLAB to implement various strategies

Course Content	TEACHING HOURS
UNIT 1: Foundations of Soft Computing and Fuzzy Logic	**14 Hrs**
Evolution of Computing: Soft Computing Constituents, From Conventional AI to Computational Intelligence: Machine Learning Basics Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions: Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzy Expert Systems, Fuzzy Decision Making	
UNIT 2: Neural Networks and Genetic Algorithms in Machine Learning	**14 Hrs**
Machine Learning Using Neural Network, Adaptive Networks, Feed forward Networks, Supervised Learning Neural Networks, Radial Basis Function Networks : Reinforcement Learning, Unsupervised Learning Neural Networks, Adaptive Resonance architectures, Advances in Neural networks Introduction to Genetic Algorithms (GA), Applications of GA in Machine Learning : Machine Learning Approach to Knowledge Acquisition	
UNIT 3: Programming for Soft Computing Applications Using Matlab/Python	**14 Hrs**
Matlab/Python Lib: Introduction to Matlab/Python, Arrays and array operations, Functions and Files, Study of neural network toolbox and fuzzy logic toolbox, Simple implementation of Artificial Neural Network and Fuzzy Logic	
UNIT 4: Advanced Topics in Soft Computing and Deep Learning	**14 Hrs**
Recent Trands in deep learning, various classifiers, neural networks and genetic algorithm. Implementation of recently proposed soft computing technique	

Textbooks

1. Jyh:Shing Roger Jang, Chuen:Tsai Sun, EijiMizutani, Neuro:Fuzzy and Soft Computing

Reference Books

1. George J. Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
2. MATLAB Toolkit Manual

COURSE OUTCOMES (CO):

CO1: Identify and describe soft computing techniques and their roles in building intelligent machines

CO2: Apply fuzzy logic and reasoning to handle uncertainty and solve various engineering problems.

CO3: Apply genetic algorithms to combinatorial optimization problems.

CO4: Evaluate and compare solutions by various soft computing approaches for a given problem

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	3	3	2	2	1	2	-	1	1	1
2	3	3	2	2	3	1	-	2	3	2	1	1
3	3	3	2	3	2	2	1	-	2	2	1	-
4	3	3	3	2	2	1	1	2	2	1	-	1

Practical Topics for Lab:

- **Fuzzy Set Exploration:** Create and visualize fuzzy sets using Python or Matlab.
- **Fuzzy Set Operations:** Implement union, intersection, and complement operations on fuzzy sets.
- **Fundamentals of Neural Networks:** Develop a simple feedforward neural network in Python.
- **Supervised Learning in Neural Networks:** Experiment with training techniques and activation functions.
- **Matlab/Python Foundations:** Work with arrays, array operations, and functions.
- **Exploring Neural Network Libraries:** Use Matlab/Python libraries for neural network implementation.
- **Using the Fuzzy Logic Toolbox:** Apply fuzzy logic functions using Matlab/Python libraries.
- **Artificial Neural Network Development:** Build and train a neural network model for basic applications.
- **Designing Fuzzy Logic Systems:** Create a fuzzy logic system to address real-world problem scenarios.
- **Deep Learning Architecture Exploration:** Experiment with convolutional and recurrent neural networks.
- **Classifier Development:** Build and compare various classifiers such as SVMs and decision trees.
- **Genetic Algorithms in Deep Learning:** Apply genetic algorithms for hyperparameter optimization in deep learning models.

COURSE TITLE: Minor Project							
Course Code:				**CSE-24203C**		**Examination Scheme**	
Total number of Lecture Hours:						**External**	**40**
						Internal	**10**
Lecture (L):	**-**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**	**Total Credits**	**2**

Program Elective III

CSE-24204XEXX

To be effective from year-2024

Program Elective IV

CSE-24205XEXX

Audit Course

CSE-24206

To be effective from year-2024

Semester III

Program Elective V

CSE-24301XEXX

To be effective from year-2024

Open Elective

CSE-24302XOE

Dissertation-I/Industrial Project

CSE-24303

To be effective from year-2024

Semester IV

To be effective from year-2024

Dissertation II

CSE-24401

Specialization: Data Science (DS)

Program Elective I

To be effective from year-2024

COURSE TITLE: Data Science									
Course Code:				**CSE-241041EDS**			**Examination Scheme**		
Total number of Lecture Hours: 56							**External**	**80**	
							Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**-**	**Tutorial (T):**	**-**		**Total Credits**	**4**	

Course Objectives

- Introduce core data science concepts and processes.
- Teach effective data collection, cleaning, and management.
- Develop skills in statistical analysis and basic machine learning.
- Enable creation of clear, impactful data visualizations

Course Content	TEACHING HOURS
UNIT 1:	**14 Hrs**
Introduction to core concepts and technologies: Introduction, Terminology, data science process, data science toolkit, Types of data, Example applications.	
UNIT 2:	**14 Hrs**
Data collection and management: Introduction, Sources of data, Data collection and APIs, Exploring and fixing data, Data storage and management, Using multiple data sources.	
UNIT 3:	**14 Hrs**
Data analysis: Introduction, Terminology and concepts, Introduction to statistics, Central tendencies and distributions, Variance, Distribution properties and arithmetic, Samples/CLT, Basic machine learning algorithms, Linear regression, SVM, Naive Bayes.	
UNIT 4:	**14 Hrs**
Data visualisation: Introduction, Types of data visualization, Data for visualisation: Data types, Data encodings, Retinal variables, Mapping variables to encodings, Visual encodings.	

Textbooks

- Doing Data Science: Straight Talk from the Frontline – Cathy O'Neil & Rachel Schutt, 2nd Ed., O'Reilly, 2023.
- Mining of Massive Datasets – Jure Leskovec, Anand Rajaraman & Jeffrey Ullman, 3rd Ed., Cambridge, 2022.
- "Data Science and Analytics" by V.K. Jain, Khanna Publishing, 1st Edition, 2020. Covers fundamental data science techniques and applications.

Reference Books

1. Data Science for Business – Foster Provost & Tom Fawcett, 2nd Ed., O'Reilly, 2021.
2. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach & Vipin Kumar, 2nd Ed., Pearson, 2020.
3. Practical Data Science with R – Nina Zumel & John Mount, 3rd Ed., Manning, 2023.

COURSE OUTCOMES (CO):

- **CO1**: Apply core data science concepts and processes.
- **CO2**: Collect, clean, and manage data from various sources.
- **CO3**: Use statistics and machine learning to analyze data.
- **CO4**: Create and interpret data visualizations effectively.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	2	-	-	3	-	-	-	-	-	-	-
2	-	3	-	2	-	-	-	3	-	-	-	-
3	-	2	-	2	-	-	-	-	-	-	-	-
4	-	-	3	-	2	-	-	-	-	-	-	-

COURSE TITLE: Distributed Systems

Course Code:			**CSE-241042EDS**			**Examination Scheme**	
Total number of Lecture Hours: 56						**External**	**80**
						Internal	**20**
Lecture(L):	**4**	**Practicasls(P):**	**0**	**Tutorial(T):**	**0**	**Total Credits**	**4**

Course Objectives

- To understand the Fundamentals of Distributed Systems
- To explore algorithms used for resource allocation, mutual exclusion, deadlock detection, and consensus in distributed systems.
- To understand the design and operation of distributed databases, replication, and consistency models.
- To explore Real-World Distributed System Architectures

Course Content	TEACHING HOURS
UNIT 1:	**- Hrs**
Introduction: Distributed data processing; DDBS; Advantages and disadvantages of DDBS; Problem areas; Overview of database and computer network concepts **Distributed Database Management System Architecture:** Transparencies in a distributed DBMS; Distributed DBMS architecture; Global directory issues.	**14**
UNIT 2:	**- Hrs**
Distributed Database Design Alternative design strategies; Distributed design issues; Fragmentation; Data allocation **Semantics Data Control** View management; Data security; Semantic Integrity Control **Query Processing Issues** Objectives of query processing; Characterization of query processors; Layers of query processing; Query decomposition; Localization of distributed data	**14**
UNIT 3:	**- Hrs**
Distributed Query Optimization Factors governing query optimization; Centralized query optimization; Ordering of fragment queries; Distributed query optimization algorithms. **Transaction Management** The transaction concept; Goals of transaction management; Characteristics of transactions; Taxonomy of transaction models	**14**
UNIT 4:	**- Hrs**
Concurrency Control Concurrency control in centralized database systems; Concurrency control in DDBSs; Distributed concurrency control algorithms; Deadlock management, Reliability issues in DDBSs. **Parallel Database Systems** Parallel architectures; parallel query processing and optimization; load balancing	**14**
Textbooks	

1. Raymond Greenlaw, H. S. S. S. G. H. M. S. R. W. A. W. M. H. S. A. D. Distributed Algorithms: An Intuitive Approach. 1st ed., Morgan Kaufmann, 2023.
2. Andrew S. Tanenbaum, Maarten Van Steen. Distributed Systems: Principles and Paradigms. 3rd ed., Pearson, 2021.
3. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. Distributed Systems: Concepts and Design. 5th ed., Pearson, 2011.
4. Distributed Database Systems, D. Bell and J. Grimson, Addison-Wesley, 1992.

Reference Books

1. Larisa Maksimov, Gregor v. H. Schreiber, T. H. A. R. M. K. S. G. T. Distributed Systems for System Architects. 2nd ed., Wiley, 2022.
2. Floyd E. Williams. The Complete Guide to Distributed Computing. 1st ed., O'Reilly Media, 2021.
3. H. J. Schneider. Distributed Computing and Internet Technologies: A Programming Approach. 1st ed., Springer, 2022.
4. Tarek S. S. Introduction to Distributed Systems. 3rd ed., Wiley, 2020.

COURSE OUTCOMES(CO):

CO1: Comprehend the Key Concepts of Distributed Systems:

CO2: Design trends in distributed systems.

CO3: Apply remote method invocation and objects.

CO4: Design and manage distributed databases, apply data consistency models

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	1	2	3	4	5	6	7	8	9	10	11	12
1	3	3	2	2	2	2	1	1	2	2	1	2
2	3	3	2	3	3	2	1	1	2	2	2	3
3	3	3	3	3	3	2	1	1	2	3	2	3
4	2	2	2	2	3	1	1	1	3	3	2	2

COURSE TITLE: Data Preparation and Analysis									
Course Code:				**CSE-241043EDS**			**Examination Scheme**		
Total number of Lecture Hours: 56							**External**	**80**	
							Internal	**20**	
Lecture (L):	**4**	**Practicals(P):**	**0**	**Tutorial (T):**	**0**		**Total Credits**	**4**	

Course Objectives

1. Introduce data gathering, cleaning, and transformation concepts.
2. Teach exploratory data analysis (EDA) skills using statistical methods.
3. Develop data visualization abilities for effective communication.
4. Familiarize with real-time data handling and scalable solutions.

Course Content	TEACHING HOURS
UNIT 1: Data Gathering and Preparation:	**14 Hrs**
Data Gathering and Preparation: Data formats, parsing and transformation, Scalability and real-time issues	
UNIT 2: Data Cleaning:	**14 Hrs**
Data Cleaning: Consistency checking, Heterogeneous and missing data, Data Transformation andsegmentation	
UNIT 3: Exploratory Analysis:	**14 Hrs**
Exploratory Analysis: Descriptive and comparative statistics, Clustering and association, Hypothesisgeneration	
UNIT 4: Visualization:	**14 Hrs**
Visualization: Designing visualizations, Time series, Geolocated data, Correlations andconnections, Hierarchies and networks, interactivity	

Textbooks

1. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python" by Peter Bruce, Andrew Bruce, and Peter Gedeck, 2nd edition, 2020, O'Reilly Media.
2. Data Science and Analytics" by V.K. Jain, 1st edition, 2020, Khanna Publishing.

Reference Books

1. "Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining" by Glenn J. Myatt, 3rd edition, 2020.

COURSE OUTCOMES (CO):

CO1: Gather and process data using transformation techniques.
CO2: Clean data, addressing inconsistencies and missing values.
CO3: Apply exploratory data analysis to identify patterns.
CO4: Create and interpret various visualizations for decision-making.

LEVEL OF CO-PO MAPPING TABLE

	POs											
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	2	–	–	3	–	–	–	–	–	–	–
2	–	3	–	2	–	–	–	3	–	–	–	–
3	–	2	–	2	–	–	–	–	–	–	–	–
4	–	–	3	–	2	–	–	–	–	–	–	–