# SEMESTER III

# MAJOR COURSE

Tobeeffectivefromyear-2024

# COURSE TITLE: Data Science

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | | **IMDAMJDS0324** | | | | **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**

- Ability to understand the fundamental concepts and techniques of data science.
- Ability to identify, analyze, and design solutions for handling large data.
- Explore the various machine learning algorithms used in the data science process.
- Acquire the skills in handling data science visualization tools for domain-specific problems.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT1:IntroductiontoDataScience** | **- Hrs** |
| Introduction to Data Science. Workflow of Data Science. Tools and Programming Languages. Data Sources and Formats. Data Science Process. Data collection methods- Data cleaning and pre-processing. Data exploration. Data modeling. | **14** |
| **UNIT2:ExploratoryDataAnalysis** | **-Hrs** |
| Data distribution analysis. Basic tools(plots, graphs and summary statistics) ofEDA. Introduction to Big Data. Steps in Big Data. Feature Engineering. Generaltechniques for handling large data, missing data. | **14** |
| **UNIT3:IntroductiontoMachineLearning** | **-Hrs** |
| Introduction to Machine Learning. Supervised and Unsupervised Machine Learning. Classification and Regression- Linear and Logistic regression. Decision trees and random forests. Clustering techniques. Dimensionality reduction. Overview of Neural Networks and Deep learning. | **14** |
| **UNIT4:ModelEvaluation** | **-Hrs** |
| Model evaluation metrics- Accuracy, Precision and Recall. Validation techniques- cross validation. Over-fitting and under-fitting. Hyper-parameter tuning. Data Visualization Techniques using R packages, Tableau. | **14** |
| **Textbooks** | |

Tobeeffectivefromyear-2024

1. Davy Cielen, Arno D. B. Meysman, Mohamed Ali. "Introducing Data Science." Manning Publications Co., 1st edition, 2016.
2. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. "An Introduction to Statistical Learning: with Applications in R." Springer, 1st edition, 2013.

**Reference Books**

1. DJ Patil, Hilary Mason, Mike Loukides. "Ethics and Data Science." O'Reilly, 1st edition, 2018.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville. "Deep Learning." MIT Press, 1st edition, 2016.
3. Joel Grus. "Data Science from Scratch: First Principles with Python." O'Reilly, 1st edition, 2015.
4. Cathy O'Neil, Rachel Schutt. "Doing Data Science: Straight Talk from the Frontline." O'Reilly, 1st edition, 2013.

**COURSEOUTCOMES(CO):**

**CO1:** Collect, clean, and pre-process data from various sources.
**CO2:** Conduct exploratory data analysis using statistical methods and data visualization techniques.
**CO3:** Build and evaluate basic predictive models using machine learning algorithms.
**CO4:** Communicate data insights effectively through written reports and visualizations.

**LEVELOFCO-POMAPPINGTABLE**

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

Tobeeffectivefromyear-2024

| COURSE TITLE: Data Structures Using C | | | | | | | |
|---|---|---|---|---|---|---|---|
| Course Code: IMDAMJDC0324 | | | | Examination Scheme | T | P | |
| Total number of Lecture Hours:42<br>Total number of Practical Hours:- | | | | External | 80 | | - |
| | | | | Internal | 20 | | - |
| Lecture(L): | 3 | Practical(P): | 1 | Tutorial(T): | 0 | Total Credits | 4 |

**Course Objectives**

1. Understand and implement linear data structures such as arrays and linked lists, including operations like insertion, deletion, and searching.
2. Master stack and queue operations, including their representations in memory and implementations using arrays and linked lists.
3. Comprehend tree structures including binary trees, binary search trees, AVL trees, and B-trees, along with their traversal techniques and applications.
4. Learn graph terminology, representations, traversal techniques, and practical applications in computer science.
5. Explore advanced data structures such as threaded binary trees and M-way search trees.

| Course Content | TEACHING HOURS |
|---|---|
| **Unit I: Linear Data Structures** | **14 Hrs** |
| Data types/objects/structures, data structures and its types, representation and implementation. Linear Data Structures: Array representation, operations, applications, and limitations of linear arrays.<br>Searching Techniques: Linear Search, Binary Search.<br>Sorting Techniques: Selection, Insertion sort, Bubble sort, Quick Sort, Merge Sort. Two-dimensional arrays, matrices, common operations of matrices, special matrices, array representation of sparse matrices.<br>Linked Lists: Representation, types, and operations on linked lists. | |
| **Unit II: Stack and Queues** | **14 Hrs** |
| Stack: Representation of stack in memory, operations on stacks, implementation of stack using arrays and linked lists, applications of stacks: Parenthesis Checker, infix to postfix procedure, evaluating expressions in postfix notation, implementation of recursion using stack.<br>Queues: Representation of queue in memory, operations on queue, implementation of queue using arrays and linked lists, circular queue and its operations, representation and implementation, deque, priority queue, heap representation of a priority queue, applications of queues. | |
| **Unit III: Tree and Graph Data 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.<br>Graphs: Terminology, graph representations, traversal techniques, operations on graphs, applications of graphs.<br>Minimum spanning trees, shortest path algorithms in graphs, Eulerian tour, Hamiltonian tour.<br>Hashing: Direct address tables, hash table, different hash functions, resolving collisions, rehashing. | |

**Textbooks**

1. Langsam, Augenstein, Tenenbaum, "Data Structures Using C and C++", 2nd Edition, 2015

**Reference Books**

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson Freed. "Fundamentals of Data Structures in C," 2nd Edition, 2018.
2. Mark Allen Weiss. "Data Structures and Algorithm Analysis in C," 3rd Edition, 2013.
3. Aho, Alfred V., Hopcroft, John E., Ullman, Jeffrey D. "Data Structures and Algorithms," 2017.
4. R. B. Patel. "Algorithms and Data Structures in C," 2020.
5. Reema Thareja. "Data Structures using C," 2nd Edition, 2016.

**Lab Manual**

1. Write a program in C to insert, delete, and update the contents of an array.
2. Write a program in C to perform various operations on matrices.
3. Write a program to add two sparse matrices.
4. Write a program to search an element using Linear Search.
5. Write a program to search an element using Binary Search.
6. Write a program to sort the elements of an array using selection sort.
7. Write a program to sort the elements of an array using bubble sort.
8. Write a program to sort the elements of an array using insertion sort.
9. Write a program to sort the elements of an array using heap sort.
10. Write a program to implement a singly linked list.
11. Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, and after a certain count of nodes in a linked list.
12. Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, and after a certain count of nodes in a linked list.
13. Write a program to reverse a linked list by changing the links in the nodes.
14. Write a program to add two polynomials represented as linked lists.
15. Write a program to multiply two polynomials represented as linked lists.
16. Write a program to implement a doubly linked list.
17. Write a program to implement different operations like adding a node at beginning, end, center, after a certain element, and after a certain count of nodes in a doubly linked list.

18. Write a program to implement different operations like deleting a node at beginning, end, center, after a certain element, and after a certain count of nodes in a doubly linked list.
19. Write a program to implement various operations on an array-based stack.
20. Write a program to implement various operations on a stack represented using a linked list.
21. Write a program to demonstrate the implementation of various operations on a linear queue represented using a linear array.
22. Write a program to demonstrate the implementation of various operations on a circular queue represented using a linear array.
23. Write a program to demonstrate the implementation of various operations on a queue represented using a linked list.
24. Write a program to create a binary tree.
25. Write a program to implement the traversal techniques of a binary tree.
26. Write a program to create a binary search tree.
27. Write a program to search for a node in a binary search tree.
28. Write a program to implement the graph using different representations.
29. Write a program to illustrate the traversal of a graph using Breadth First Search and Depth First Search.
30. Write a program to find the shortest path in a graph using Floyd and Warshall's Algorithm.

**COURSEOUTCOMES(CO):**

**CO1:** Students will be able to implement and manipulate linear data structures such as arrays, linked lists, and matrices, including operations like insertion, deletion, and traversal.

**CO2:** Students will demonstrate proficiency in implementing and applying advanced data structures such as stacks, queues, trees (binary trees, AVL trees), graphs, and various heaps (binomial heaps, leftist heaps) to solve complex problems.

**CO3:** Students will develop analytical and problem-solving skills by applying appropriate data structures and algorithms to solve practical problems related to data storage, retrieval, and manipulation in computer science applications.

**LEVELOFCO-POMAPPINGTABLE**

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

Tobeeffectivefromyear-2024

# MINORCOURSE

Tobeeffectivefromyear2024

## COURSE TITLE: Computer Networks

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code: IMDAMNCN0324** | | | | | | **Examination Scheme** | |
| **Total number of Lecture Hours: 56** | | | | | | **External** | **80** |
| | | | | | | **Internal** | **20** |
| **Lecture(L):** | **4** | **Practicals(P):** | **-** | **Tutorial(T):** | **-** | **Total Credits** | **4** |

**Course Objectives**

- To understand the fundamental goals, applications, and topologies of network architectures including LAN, MAN, and WAN.
- To explore the OSI and TCP/IP models, their comparison, and the role of protocol layers and service models in network communication.
- To learn internetworking concepts, including connection-oriented and connection-less approaches, IP addressing, and the structure of IP headers.
- To analyze internet control protocols and principles of routing, congestion control, and reliable data transfer.
- To gain practical knowledge in transport layer protocols (UDP and TCP), and the application of socket APIs for network programming.
- To develop skills in evaluating and implementing network design and troubleshooting techniques in real-world scenarios.

| **Course Content** | **TEACHING HOURS** |
|---|---|
| **Unit 1: Introduction to Networking and Network Architectures** | **14 Hrs** |
| Goals and applications of networks, Network topologies: LAN, MAN, and WAN architectures, Concept of WAN subnet, Overview of existing networks, OSI Reference Model Architecture and TCP/IP Model, and their comparison, Protocol layers and service models, OSI and Internet protocols | |
| **Unit 2:** Internetworking and IP Addressing | **14Hrs** |
| Internetworking concepts and architectural models, Connection-oriented and connection-less approaches, Switching techniques: Circuit switching, packet switching, and virtual circuit switching, Concept of autonomous systems and internetwork routing, IP addressing: Classful IP addresses, subnetting, and subnet addressing, IP multicasting, Internet Protocol (IP): Connectionless delivery of datagrams (MTU, fragmentation, reassembly), IP header structure | |
| **UNIT 3:** Internet Control Protocols and Routing | **14hrs** |
| Internet control protocols: ICMP, ARP, and RARP, Concepts of delay, security, and Quality of Service (QoS), Reliable data transfer and Stop-and-Wait evaluation, Principles of congestion control, Principles of routing: Link-state and distance vector routing, Routing algorithms: Inter-domain and intra-domain routing (RIP, OSPF, BGP), Classless Inter-Domain Routing (CIDR) | |

| UNIT 4 : Transport Layer and Network Programming | 14hrs |
|---|---|
| Transport layer protocols: UDP and TCP concepts, TCP and UDP semantics and syntax, Reliable transmission: Flow control and error control mechanisms, Principles of congestion avoidance, Socket API for network programming, Network programming concepts: Sockets and application layer interactions | |

**Textbooks**

- Andrew Tanenbaum, **"Computer Networks"**, 5th Edition, Pearson, 2010.
- Behrouz A. Forouzan, **"Data Communication and Networking"**, 6th Edition, McGraw-Hill Education, 2017.
- William Stallings, **"Computer Networking with Internet Protocols and Technology"**, 1st Edition, Pearson, 2018.

**ReferenceBooks**

- Douglas Comer, **"Internetworking with TCP/IP, Volume 1"**, 6th Edition, Pearson, 2014.
- W. Richard Stevens, **"UNIX Network Programming"**, 3rd Edition, Pearson, 2003.
- Mark A. Miller, **"IP Fundamentals"**, 2nd Edition, Pearson, 2013.
- Douglas Comer, **"Client-Server Programming with TCP/IP, Volume 3"**, 2nd Edition, Pearson, 2000.

**COURSEOUTCOMES(CO):**

**CO1:** Ability to describe and apply network topologies and architectures, including LAN, MAN, and WAN, and understand their goals and applications.

**CO2:** Proficiency in explaining and comparing OSI and TCP/IP models, including protocol layers and service models.

**CO3:** Competence in applying internetworking concepts, addressing techniques, and understanding IP header structures.

**CO4:** Skill in utilizing internet control protocols, principles of routing, and managing congestion control and reliable data transfer.

**CO5:** Mastery in using transport layer protocols and socket APIs for effective network programming and communication.

**CO6:** Capability to analyze and troubleshoot network design issues, applying theoretical knowledge to practical network scenarios.of generative models using appropriate metrics.

**LEVELOFCO-PO MAPPINGTABLE**

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

# ABILITY ENHANCEMENT COURSES (AEC)

**One 3-Credit AEC Course to be opted from**

**centrally organized Basket of courses for**

**FYIMP under NEP 2020**

**IMDAAEXXXX23**

# MULTI DISCIPLINARY COURSES (MDC)

**One 3-Credit MDC Course to be opted from**

**centrally organized Basket of courses for**

**FYIMP under NEP 2020**

**IMDAMDXX0023**

# SKILL ENHANCEMENT COURSE (SEC)

## COURSE TITLE: Matlab

| Course Code: IMDASEML0324 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of Lecture Hours:28<br>Total number of Practical Hours:- | | | | | | External | 40 | - |
| | | | | | | Internal | 10 | - |
| Lecture(L): | 2 | Practical(P): | 0 | Tutorial(T): | 0 | Total Credits | 2 | |

**Course Objectives**

- Understand the fundamentals of MATLAB programming, including the workspace, variables, arrays, and logical operations.
- Develop proficiency in controlling program flow with conditional statements and loops.
- Gain the ability to create and use MATLAB functions, manage files, and handle data input/output efficiently.
- Master the basics of plotting and data visualization, including customizing plot features and saving graphical output.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT1:MATLAB Basics and Branching/Looping Constructs** | **Hrs.** |
| Overview of MATLAB features and workspace. Variables and Arrays: Working with multidimensional arrays, subarrays, scalar, and matrix operations. Displaying output, logical data types.<br><br>Branching: Conditional statements using the if, switch, and try/catch constructs. Logical Arrays: Utilizing logical arrays to simulate if/else constructs.<br><br>Loops: Implementing while and for loops, vectorization as a faster alternative to loops, and understanding the Just-In-Time (JIT) compiler. Handling loops with break and continue statements and nesting loops. | 14 |
| **UNIT2:Functions, File Handling, and Plotting** | **Hrs.** |
| Introduction to MATLAB functions and variable passing: Pass-by-value scheme. Working with optional arguments and preserving data between function calls. Exploring built-in MATLAB functions for sorting, random number generation, and more.<br><br>File Handling: Understanding file processing in MATLAB. Opening and closing files using fopen and fclose functions. Handling binary I/O functions with fwrite and fread. Managing file positioning and using the textscan function.<br><br>Plotting in MATLAB: Introduction to plotting with simple xy plots. Customizing plots with line color, style, markers, and legends. Creating multiple plots, using logarithmic scales, controlling axes limits, and generating polar plots. Advanced two-dimensional plotting and working with two-dimensional arrays in the plot function. | 14 |

**Textbooks**


1. Moler, Cleve. "MATLAB: An Introduction with Applications." 5th ed., Chapman and Hall/CRC, 2021.
2. Attaway, Stormy. "MATLAB: A Practical Introduction to Programming and Problem Solving." 3rd ed., Elsevier, 2021.
3. Higham, Desmond J., and Nicholas J.Higham."MATLAB Guide".2nded.,SIAM,2016.


**Reference Books**


1. Schaffer, J. D., and E. J. McGowan. "MATLAB for Engineers." 3rd ed., Cengage Learning, 2019.
2. Moore, J. J. "MATLAB for Dummies." 2nd ed., Wiley, 2019.
3. Demuth, Howard, et al. "MATLAB Neural Network Toolbox User's Guide." 1st ed., The MathWorks, 2021.


**COURSE OUTCOMES(CO):**

**CO1:** Understand the fundamentals of MATLAB programming, including the workspace, variables, arrays, and logical operations.

**CO2:** Develop proficiency in controlling program flow with conditional statements and loops.

**CO3:** Gain the ability to create and use MATLAB functions, manage files, and handle data input/output efficiently.

**CO4:** Master the basics of plotting and data visualization, including customizing plot features and saving graphical output.

**LEVELOFCO-POMAPPINGTABLE**

| | Pos | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |

# SEMESTER IV

# MAJOR COURSE

Tobeeffectivefromyear-2024

# COURSE TITLE: Algorithms Design and Analysis

| Course Code: IMDAMJAD0424 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of Lecture Hours:56<br>Total number of Practical Hours:- | | | | | | External | 80 | - |
| | | | | | | Internal | 20 | - |
| Lecture(L): | 4 | Practical(P): | - | Tutorial(T): | - | Total Credits | | 4 |

**Course Objectives**

- Evaluate algorithms using asymptotic notations and complexity analysis techniques.
- Implement and analyze the efficiency of various search and sorting algorithms.
- Apply divide-and-conquer, greedy, backtracking, and dynamic programming strategies to solve complex problems.
- Design and evaluate approximation algorithms, including understanding their performance guarantees and specific applications to problems like TSP and Vertex Cover.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT1:** FundamentalsofAlgorithms | **14Hrs** |
| Introduction to Algorithms, Analysis of Algorithms, Growth of functions, Asymptotic Notations (Big-O, Big-Omega, Big-Theta), Complexity Analysis Techniques (Substitution method, Recursion Tree),Masters Theorem. Tower of Hanoi problem and its complexity. | |
| **UNIT2:** SearchandSortingAlgorithms | **14 Hrs** |
| Search Algorithms (Linear Search, Binary Search), Sorting Algorithms (Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort). Time complexity analysis of searching and Sorting algorithms. | |
| **UNIT3:** AdvancedProblem-SolvingStrategies | **14 Hrs** |
| Introduction to Divide and Conquer strategy, Greedy method, Knapsack problem, Introduction to Backtracking Strategy, 8-Queens problem, Dynamic programming Strategy, All pair shortest path problem, Branch and Bound Strategy. | |
| **UNIT4:** Complexity Classes and Approximation Algorithms | **14 Hrs** |

Understanding complexity classes, including P, NP, NP-Complete, and NP-Hard problems. Basic concepts and the need for approximation in solving NP-hard problems. Techniques for designing greedy approximation algorithms, such as the Fractional Knapsack problem.

Approximation Ratios and Performance Guarantees: Understanding how to measure the performance of approximation algorithms and derive approximation ratios. Overview of specific problems and their approximation algorithms, such as the Traveling Salesman Problem (TSP), Vertex Cover, and Set Cover.

**Textbooks**


1. Cormen, Thomas H., et al. *Introduction to Algorithms.* 4th ed., MIT Press, 2022.
2. Kleinberg, Jon, and Éva Tardos. *Algorithm Design.* 2nd ed., Pearson, 2016.
3. Dasgupta, Sanjoy, Christos H. Papadimitriou, and Umesh V. Vazirani. *Algorithms.* 2nd ed., McGraw-Hill, 2020.
4. Horowitz, Sahni, and Rajasekaran. *Fundamentals of Computer Algorithms,* Galgotia Publications.


**Reference Books**


1. Sedgewick, Robert, and Kevin Wayne. *Algorithms.* 4th ed., Addison-Wesley, 2021.
2. Skiena, Steven S. *The Algorithm Design Manual.* 3rd ed., Springer, 2022.
3. Levitin, Anany. *Introduction to the Design and Analysis of Algorithms.* 3rd ed., Pearson, 2018.


**COURSEOUTCOMES(CO):**

**CO1:** Students will use asymptotic notations and complexity analysis techniques to evaluate and compare algorithm efficiency.
**CO2:** Students will implement search and sorting algorithms and assess their time complexity.
**CO3:** Students will apply divide-and-conquer, greedy, backtracking, and dynamic programming strategies to solve complex computational problems.
**CO4:** Students will design and evaluate approximation algorithms for NP-hard problems and measure their performance using approximation ratios.

**LEVELOFCO-POMAPPINGTABLE**

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

Tobeeffectivefromyear-2024

| COURSE TITLE: Linear Algebra | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Course Code:** | **IMDAMJLA0424** | | | | | **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. Understand the fundamental concepts of linear algebra and its application in computer science.
2. Learn matrix operations, vector spaces, and their use in solving systems of linear equations.
3. Explore eigenvalues, eigenvectors, and their relevance in data science and machine learning.
4. Apply linear transformations and decompositions in practical scenarios, such as graphics and computer vision.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT 1:Introduction to Linear Algebra:** | **-Hrs** |
| **Scalars, Vectors, and Matrices**: Definitions, notation, and basic, operations. Matrix operations: Addition, scalar multiplication, matrix multiplication, and transposition. Systems of linear equations: Row reduction, echelon forms, and Gaussian elimination. **Determinants and Matrix Inverses**: Definition and properties of determinants, Computation of determinants and matrix inverses. Application: Solving linear systems using inverses | **14** |
| **UNIT 2: Vector Spaces and Subspaces** | **-Hrs** |
| **Definitions**: Formal definitions of vector spaces and subspaces. Linear Combinations and Span: Understanding the span of vectors and its significance. **Basis and Dimension**: Defining the basis of a vector space and exploring the concept of dimension. Rank-Nullity Theorem. Row space, column space, and null space of a matrix. **Linear Independence**: Understanding linear independence and dependence in vectors, with real-world applications in error detection, coding theory, and computer graphics | **14** |
| **UNIT 3:Eigenvalues, Eigenvectors, and Diagonalization** | **-Hrs** |
| **Eigenvalues and Eigenvectors**: Definitions and methods to calculate eigenvalues and eigenvectors. Characteristic polynomial and its role.<br>**Diagonalization**: Criteria for diagonalizing matrices. Application: Simplifying computations using diagonal matrices.<br>**Applications in Machine Learning and Data Science**: Introduction to Principal Component Analysis (PCA) and its use in dimensionality reduction. Role of eigenvalues in Markov models and Google's PageRank algorithm. | **14** |
| **UNIT 4: Linear Transformations and Matrix Factorization** | **- Hrs** |
| **Linear Transformations**: Understanding the concept of mapping between vector spaces, matrix representation of linear transformations.<br>Kernel and image of linear transformations.<br>**Matrix Factorizations:** LU decomposition for solving systems of equations.<br>QR decomposition for solving least squares problems.<br>Singular Value Decomposition (SVD)<br>**Applications in Computer Science:** Applications in graphics transformations, machine learning algorithms (e.g., neural networks, SVD in recommendation systems), and robotics. | **14** |

**Textbooks**

1. "Linear Algebra and Its Applications" by David C. Lay, 6th Edition (2021), Pearson.
2. "Introduction to Linear Algebra" by Gilbert Strang, 5th Edition (2023), Wellesley-Cambridge Press.
3. "Elementary Linear Algebra" by Howard Anton and Chris Rorres, 12th Edition (2022), Wiley.
4. "Linear Algebra" by S. Kumaresan, 2nd Edition (2017), Prentice-Hall of India.

**Reference Books**

1. "Schaum's Outline of Linear Algebra" by Seymour Lipschutz and Marc Lipson, 6th Edition (2020), McGraw-Hill Education.
2. "Advanced Engineering Mathematics" by Erwin Kreyszig, 11th Edition (2020), Wiley India.

**COURSEOUTCOMES(CO):**

**CO1:** Understand vectors, matrices, and systems of linear equations.
**CO2:** Analyze vector spaces and subspaces.
**CO3:** Solve problems using matrix operations and eigenvalues.
**CO4:** Apply linear algebra techniques to real-world scenarios.

**LEVELOFCO-PO MAPPINGTABLE**

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

Tobeeffectivefromyear-2024

## COURSE TITLE: Artificial Intelligence

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Course Code:** | | **IMDAMJAI0424** | | | | **Examination Scheme** | **T** | **P** |
| **TotalnumberofLectureHours:42**<br>**TotalnumberofPracticalHours:28** | | | | | | **External** | **80** | **-** |
| | | | | | | **Internal** | **20** | **-** |
| **Lecture(L):** | **3** | **Practical(P):** | **1** | **Tutorial(T):** | **0** | **Total Credits** | **4** | |

**Course Objective:**

- To develop a solid understanding of the basic principles and history of artificial intelligence.
- Learn how to represent and organize knowledge for intelligent systems.
- Understand and apply reasoning methods for decision-making and problem-solving.
- To implement and apply algorithms to solve complex problems.

| **Course Content** | **TEACHING HOURS** |
|---|---|
| **UNIT1:IntroductiontoArtificialIntelligence** | **-14Hrs** |
| Introduction and historical perspective, Artificial Intelligence-disciplines and applications, Types of AI, Theories of intelligence, Detecting and Measuring Intelligence, Turing Test, Future of AI Intelligent agent:Agent and Environments, nature of Environments, structure of Agent, Concept of Rationality<br>SearchStrategies:Uninformedsearchstrategies-Breadthfirstsearch,uniform costsearch,Depthfirstsearch,DepthLimitedandIterativeDeepeningSearch. | |
| **UNIT2:SearchAlgorithms** | **-14Hrs** |
| Search with partial information (Heuristic Search), greedy best-first search, A* search, heuristic functions.<br><br>Local Search Algorithm: Hill climbing, gradient descent, genetic algorithm.<br><br>Game Playing: Game theory, adversarial search, minimax algorithm, optimal decisions in multiplayer games, alpha-beta pruning.<br><br>Knowledge Representation: Propositional logic, first-order logic, inference in first-order logic, propositional versus first-order logic. | |

Tobeeffectivefromyear-2024

| UNIT3: Expert System and Fuzzy Logic | -14Hrs |
|---|---|
| Expert systems, forward chaining, backward chaining, conflict resolution, knowledge representation, representing uncertain knowledge, and reasoning with uncertain information.<br><br>Fuzzy logic, fuzzification, fuzzy sets, operations on fuzzy sets, hedges, reasoning in fuzzy logic—Mamdani inference. | |

**LAB MANUAL**

1. Given a real-life situation, identify the characteristics of the environment, identify the percepts available to the agent, identify the actions that the agent can execute, suggest the performance measures to evaluate the agent, and recommend the architecture of the desired intelligent agent.
2. Undertake a comparative study on popular programming languages used in AI.
3. Translate commonly used English sentences into their equivalent logical expressions.
4. Given a problem description, formulate it in terms of a state space search problem. Analyze the given problem and identify the most suitable search strategy for the problem. Make an analysis of the properties of proposed algorithms in terms of:
    - Time complexity
    - Space complexity
    - Optimality
5. Write a program to implement Breadth First Search.
6. Write a program to implement Depth First Search.
7. Write a program to implement Hill Climbing Algorithm.
8. Write a program to implement A* Algorithm.
9. Write a program to implement Tic-Tac-Toe game.
10. Write a program to implement 8-Puzzle problem.
11. Write a program to implement Water-Jug problem.
12. Write a program to implement Alpha-Beta Pruning using Python.
13. Write a program to implement 8-Queens Problem.
14. Build a fuzzy inference system for the Tipping Problem. Given two sets of numbers between 0 and 5 (where 0 is for very poor and 5 is for excellent) that respectively represent quality of service and quality of food at the restaurant, what should the tip be?

15. Solve 2-input 1-output project risk prediction problem using Mamdani Inference. Make necessary assumptions.

**Textbooks**


- "Artificial Intelligence: A Guide to Intelligent Systems" by Michael Negnevitsky, Latest Edition, 2020.
- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig, 4th Edition, 2020.
- "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell, Latest Edition, 2019.

**Reference Books**


- "Artificial Intelligence" by Elaine Rich, Kevin Knight, and Shivashankar B. Nair, 4th Edition, 2021.
- "Artificial Intelligence: Foundations of Computational Agents" by Michael Wooldridge, 1st Edition, 2021.


**COURSEOUTCOMES(CO):**

**CO1:** Identify and discuss various applications of AI across different domains and their impacts.

**CO2:** Develop and implement knowledge-based systems and expert systems for decision-making and problem-solving.

**CO3:** Utilize optimization techniques to tackle complex issues.

**CO4:** Implement and evaluate informed and uninformed search algorithms to solve problem-solving tasks.

**LEVELOFCO-POMAPPINGTABLE**

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

# COURSE TITLE: DBMS

| Course Code: IMDAMJDB0424 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of Lecture Hours:56<br>Total number of Practical Hours:- | | | | | | External | 80 | - |
| | | | | | | Internal | 20 | - |
| Lecture(L): | 4 | Practical(P): | - | Tutorial(T): | - | Total Credits | | 4 |

**Course Objectives**

- Understand the purpose, components, and applications of database systems and their role in modern data management.
- Apply data integrity rules and normalization techniques to design and optimize relational databases.
- Utilize SQL and NoSQL query languages to perform data manipulation, querying, and management tasks effectively.
- Implement transaction management and concurrency control methods to ensure data consistency and reliability in multi-user environments.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT1:IntroductiontoDatabases** | **14Hrs** |
| Introduction to Databases: Purpose of Database Systems, Components of DataBase Systems, Applications of Database Management Systems, Three Tier Database Management System Architecture, Data Independence, Database Schema, Instance, Data Modelling, Entity Relationship Model, Relational Model. | |
| **UNIT2:RelationalDatabaseDesign** | **14Hrs** |
| Data integrity Rules, Functional Dependency, Normalization, First, Second, Third Normal Forms, BCNF, Multi-valued Dependencies, Pit-falls in relational Database design, De-normalization. | |
| **UNIT3:QueryLanguage** | **14Hrs** |
| Relational Algebra, Introduction to Data Definition Language, Data Manipulation Language, Data Control and Transaction Control Language, Integrity Constraints, Database Keys, SQL basic Operations, Aggregate Functions, SQL Joins, Views, Subqueries, Introduction to NoSQL Query Languages: MongoDB Query Language, Cassandra Query Language (CQL) | |

| UNIT4:TransactionmanagementandConcurrencycontrol | 14Hrs |
|---|---|
| Transaction System Concepts, Desirable properties of transactions: ACID properties, Schedules, serializability and concurrency control, Lock based concurrency control (2PL, Deadlocks), Time stamping methods, optimistic methods, database recovery management. | |

**Textbooks**


1. Elmasri, Ramez, and Shamkant B. Navathe. *Fundamentals of Database Systems*. 7th ed., Pearson, 2015.
2. Silberschatz, Abraham, Henry Korth, and S. Sudarshan. *Database System Concepts*. 7th ed., McGraw-Hill, 2020.
3. Korth, Henry F., and Abraham Silberschatz. *Database System Concepts*. 7th ed., McGraw-Hill Education, 2020.


**ReferenceBooks**


1. Date, C. J. *An Introduction to Database Systems*. 8th ed., Pearson, 2003.
2. Ramakrishnan, Raghu, and Johannes Gehrke. *Database Management Systems*. 3rd ed., McGraw-Hill, 2002.
3. Rob, Peter, and Carlos Coronel. *Database Systems: Design, Implementation, & Management*. 13th ed., Cengage Learning, 2018.


**COURSE OUTCOMES (CO)**

**CO1:** Students will be able to explain the fundamental components of a database system and their functions in managing and organizing data.

**CO2:** Students will demonstrate the ability to apply normalization techniques to create efficient and well-structured relational database schemas.

**CO3:** Students will write and execute complex SQL queries and NoSQL commands to manipulate and retrieve data from various database systems.

**CO4:** Students will apply transaction management principles and concurrency control methods to handle data integrity and synchronization in a database system.

# MINOR COURSE

To be effective from year-2024

# COURSE TITLE: Software Engineering

| Course Code: IMDAMNSE0424 | | | | | | Examination Scheme | T | P |
|---|---|---|---|---|---|---|---|---|
| Total number of Lecture Hours: 56<br>Total number of Practical Hours: - | | | | | | External | 80 | - |
| | | | | | | Internal | 20 | - |
| Lecture(L): | 4 | Practical (P): | - | Tutorial(T): | - | Total Credits | | 4 |

**Course Objectives**

- **Understand software engineering fundamentals**: Grasp the nature, scope, and challenges of software development.
- **Apply software development models**: Identify and use models like Waterfall, Agile, Spiral, and RAD.
- **Master software design principles**: Learn design methodologies and use UML for modeling.
- **Evaluate software quality**: Conduct software testing and apply quality assurance practices.

| Course Content | TEACHING HOURS |
|---|---|
| **UNIT 1:** Software Engineering Fundamentals and Process Models | **14 Hrs** |
| Software Engineering Fundamentals: Concept and Nature of Software, Software Development Challenges, Software Scope, Software Engineering Discipiline, Software Process Models: Waterfall, Evolutionary, Spiral, Prototyping, RAD, Incremental and Iterative models, Agile Software Development. | |
| **UNIT 2:** Software Process Measurement and Requirements Development | **14 Hrs** |
| Software Process and Project Measurement: Measures, Metrics and Indicators, Size -Oriented Metrics vs. Function - Oriented Metrics, Capability Maturity Model Integration (CMMI). COCOMO Model. Requirements Development Methodology, Requirement Specification, Documenting Business Requirements, Defining user requirements, Validating requirements, Reviews, Walkthroughs and inspections, Requirement Modelling. | |

| **UNIT 3:** Design Engineering and Object-Oriented Design | **14 Hrs** |
|---|---|
| Basics of Design Engineering, Function oriented design, Design principles, Coupling and Cohesion, Structured Design Methodology, Object-Oriented Design - Design Concepts, Design Methodology, Object-oriented analysis and design modeling using Unified Modeling Language (UML), Dynamic & Functional Modeling. | |
| **UNIT 4:** Software Testing, Verification, and Quality Assurance | **14 Hrs** |
| Software Verification and Validation, Software Testing Concepts , Different types of testing: Unit Testing , Integration and System Testing ,Static and Dynamic Testing, Functional and non-functional Testing, Other Specialized Testing, Software Quality Concepts, Software Configuration Management (CM) ,Software Quality Assurance (SQA). | |

**Textbooks**

1. Pfleeger and Atlee, Software Engineering: Theory and Practice, 4th Edition, Pearson, 2010.

**Reference Books**

1. Sommmerville, Ian - Software Engineering. Pearson , 9/e , 2011.
2. Pankaj Jalote - An Integrated approach to Software Engineering, Narosa Publication.
3. Software Engineering: Principles and practice, 3rd Edition, Hans Van Vliet, Wiley.
4. James F. Peters Software Engineering – An Engineering Approach, Wiley& Sons.
5. Roger Pressman, Software Engineering: A Practitioners Approach",McGraw-Hill Publications.

**COURSEOUTCOMES(CO):**

**CO1:** Students will apply appropriate software process models to various project scenarios.

**CO2:** Students will use industry-standard metrics to evaluate software processes and project performance.

**CO3:** Students will design cohesive software systems using function-oriented and object-oriented methodologies.

**CO4:** Students will conduct software testing and implement quality assurance practices effectively.

**LEVELOFCO-PO MAPPINGTABLE**

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