**Mukesh Patel School of Technology Management & Engineering**

**School of Technology Management and Engineering**

**Course Policy Document**

**Course Name - (Code):** Data Structures and Algorithms (DSA) – (702CO1C001)

|  |  |
| --- | --- |
| **Program and Semester:** B Tech CSBS – Sem II  B Tech and MBA Tech (All programs except Civil and  Mechanical) – Sem III  B Tech Computer Science and Engineering (Data Science) – Sem III  B Tech Integrated Computer, – Sem VI | **Pre-requisite Course:** Programming for Problem Solving |
| **Academic Year:** 2024-25 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | L | T | P | C | H | | 3 | 0 | 2 | 4 |  |   **Credit Details:** |
| **Name of Course Faculty:** Dr. Deepti Reddy Patil  **Program:** B Tech(CE), Mumbai Campus  **Contact Details:** [deepti.reddy@nmims.edu](mailto:deepti.reddy@nmims.edu)  **Office Hours:** | **Faculty associated with the course: -**  **1.**  **2.** |
| **Pre-Course Activity:**  Using GenAI, explore the following:   1. Importance of DSA course in industry? 2. Practice C/C++ programs. | |
| **Course link:** | |

1. **Introduction to the Course**
   1. **Importance of the Course**
      1. **Domain** Relevance**:**

This course imparts knowledge of data structures and algorithms to identify and implement appropriate data structures and determine the computational complexity of the given application.

* + 1. **Industry** Relevance**:**

This course emphasizes on implementation of linear and non-linear data structures and its use in real life applications. The data structures learned in this course would be useful in understanding and implementing concepts of some courses of higher semesters. Essentially, it is a prerequisite / useful for important courses such as operating system, computer networks, database management systems, artificial intelligence and machine learning. This course also emphasizes learning different searching and sorting algorithms with its analysis.

Data Structures and algorithms are relevant in various fields like machine learning, Blockchain, big data, web development, Computer Networks, compiler construction, etc.

* 1. **Objectives of the Course:**
     1. **imparts knowledge of data structures and algorithms so as to identify and implement appropriate data structure** Understand problem statements and solve those using basic programming constructs
     2. **Determine the computational complexity of the given application**

1. **Course Outcomes (CO), Mapping with Program Outcomes (PO), and Program Specific Outcomes (PSO)**
   1. **Course Outcomes**
      1. **CO1:** Understand the concept of data structures and computational complexity
      2. **CO2:** Identify and implement appropriate linear data structure for the given problem.
      3. **CO3:** Identify and implement appropriate non-linear data structure for the given problem.
      4. **CO4:** Differentiate various searching and sorting algorithms
   2. **Program Outcomes(PO) the course contributed to:**
      1. **PO-1:** Apply knowledge of mathematics, science and engineering fundamentals to Computer Engineering problems.
      2. **PO-2:**  Identify, formulate, research literature and analyze problems using first principles of mathematics and computing.
      3. **PO-3:** Design and develop IT solutions to complex engineering problems by considering public health, safety, environmental and cultural issues.
      4. **PO-4:** Use research-based knowledge including design of experiments, analysis and interpretation of data to solve complex computer engineering problems.
      5. **PO-5:** Select or create and apply modern IT tools and techniques to predict and model complex engineering activities.
      6. **PO-6:** Apply reasoning using contextual knowledge on contemporary issues and consequent responsibilities relevant to professional engineering practice.
      7. **PO-7:**  Apply the broad knowledge necessary to understand the impact of engineering solutions on sustainable development and environment
      8. **PO-8:** Apply ethical principles and commit to professional ethics as well as norms of computer engineering practice.
      9. **PO-9:** Function effectively as an individual and a team member in multidisciplinary settings.
      10. **PO-10:** Communicate effectively on complex engineering activities with the engineering community and society at large.
      11. **PO-11:** Demonstrate knowledge and understanding of computer engineering and management principles to apply for project management in multi-disciplinary environment.
      12. **PO-12:** Engage in independent and lifelong learning to adapt technological change.
   3. **Program Specific Outcomes (PSO):** 
      1. **PSO-1:**
      2. **PSO-2:**

**CO-PO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | 3 | 1 |  |  |  |  |  |  |  |  |  |  |
| CO2 | 3 | 3 | 2 |  |  |  |  |  | 2 | 2 | 2 | 3 |
| CO3 | 3 | 3 | 2 |  |  |  |  |  | 2 | 2 | 2 | 3 |
| CO4 | 3 | 3 | 2 |  |  |  |  |  | 2 | 2 | 2 | 3 |

***Mapping Levels: 1- High, 2-Medium, 3-Low***

* 1. **Student Outcomes (SO) (For ABET accredited Programs):**
     1. **SO-1:** Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
     2. **SO-2:** Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
     3. **SO-3:** Communicate effectively in a variety of professional contexts.
     4. **SO-**4: Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
     5. **SO-**5: Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
     6. **SO-**6: Identify and analyze user needs and to take them into account in the selection, creation, integration, evaluation, and administration of computing-based systems.

1. **Teaching-learning methodology** 
   1. **Instruction Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lecture No.** | **Topic** | **Teaching Method \*** | **Blooms Level** | **Resources** | **COs mapped** | **Assessment and Evaluation** |
| **Unit 1: Introduction** | | | | | | |
| 1 | Introduction to data structure and its importance, | PPT, Whiteboard | UNDERSTAND | TB2, Chap no 2, Page no.43 | CO1 | Class Test-1 and TEE |
| 2 | Classification of data structures, Basic operations, | UNDERSTAND | TB2, Chap no 2, Page no.45-49 | CO1 |
| 3 | Abstract data type, Performance analysis- time and space complexity, | UNDERSTAND | TB2, Chap no 2, Page no.50, 54 | CO1 |
| 4 | Asymptotic Notations. | UNDERSTAND | TB2, Chap no 2, Page no.57-61 | CO1 |
| **Unit 2: Linear Data Structure I** | | | | | | |
| 5 | Representation of arrays in memory, Operations on arrays -Traversal, Insertion | PPT, Whiteboard | APPLY | TB2, Chap no 3 Page no.66 | CO2 | Lab Submission and Class Test-1 and TEE |
| 6 | Introduction to Stacks, Operations on Stacks, | APPLY | TB2, Chap no 7 Page no.219 | CO2 |
| 7 | Applications of stacks -  Expression conversion and evaluation (Polish notation), | APPLY | TB2, Chap no 7 Page no.230 | CO2 |
| 8 | Balanced parenthesis  checker, | APPLY | TB2, Chap no 7 Page no.231 | CO2 |
| 9 | Recursion, | APPLY | TB2, Chap no 7 Page no.243 | CO2 |
| 10 | Introduction to Queue,  Operation on Queues, | APPLY | TB2, Chap no 8 Page no.253 | CO2 |
| 11 | Linear queue | APPLY | TB2, Chap no 8 Page no.253 | CO2 | Lab Submission and Class Test-1 and TEE |
| 12 | Circular queue | APPLY | TB2, Chap no 8 Page no.260 | CO2 |
| 13 | Priority queue |  | APPLY | TB2, Chap no 8 Page no.268 | CO2 |  |
| 14 | Application of Queues |  | APPLY | TB2, Chap no 8 Page no.275 | CO2 |  |
| **Unit 3: Linear Data Structure II** | | | | | | |
| 15 | Introduction to linked list, Representation of linked list in memory, | PPT, Whiteboard | APPLY | TB2, Chap 6, Page 162-166 | CO3 | Lab Submission and Class Test-2 and TEE |
| 16 | Singly linked  list and its operations, | APPLY | TB2, Chap 6, Page 167-180 | CO3 | - |
| 17 | Singly linked  list and its operations, | APPLY | TB2, Chap 6, Page 167-180 | CO3 | Lab Submission and Class Test-2 and TEE |
| 18 | Introduction to Doubly Linked list (Algorithms) | APPLY | TB2, Chap 6, Page 188-194 | CO3 |
| 19 | Linked list representation of Stack and Queues, |  | APPLY | TB2, Chap 6, Page 167-180 | CO3 |  |
| 20 | Linked list representation of Stack and Queues, |  | APPLY | TB2, Chap 6, Page 224-227, 256-260 | CO3 |  |
| 21 | Applications of linked list – Polynomial  Addition |  | APPLY | TB2, Chap 6, Page 211-215 | CO3 |  |
| **Unit 4: Non-Linear Data Structures - I** | | | | | | |
| 22 | Introduction, Binary tree terminologies, | PPT, Whiteboard | Understand | TB1 Chapter no 7, Page number 7.1 | CO3 | Lab Submission and Class Test-2 and TEE |
| 23 | Representation of Binary trees in memory | Understand | TB1 Chapter no 7, Page number 7.5 | CO3 | Lab Submission and Class Test-2 and TEE |
| 24 | Binary Tree traversal algorithms | APPLY | TB1 Chapter no 7, Page number 7.13 | CO3 | Lab Submission and Class Test-2 and TEE |
| 25 | Construction of Binary Tree from traversals, | APPLY | TB1 Chapter no 7, Page number 7.77 | CO3 | - |
| 26 | Binary Search Tree: Insertion, |  | APPLY | TB1 Chapter no 7, Page number 7.25,7.32 | CO3 |  |
| 27 | Binary Search Tree: Deletion, |  | APPLY | TB1 Chapter no 7, Page number 7.32 | CO3 |  |
| 28 | Solving numerical on Binary Search Tree |  | APPLY | TB1 Chapter no 7, Page number 7.77 | CO3 |  |
| 29 | Applications of tree data  structure: Expression trees |  | APPLY | TB1 Chapter no 7, Page number 7.12 | CO3 |  |
| 30 | Huffman trees |  | APPLY | TB1 Chapter no 7, Page number 7.66 | CO3 |  |
| 31 | Solving numerical on Trees |  | APPLY | TB1 Chapter no 7, Page number 7.92 | CO3 |  |
| **Unit 5: Non- Linear Data Structures - II** | | | | | | |
| 32 | Introduction, Graph theory terminology, | PPT, Whiteboard | Understand | TB2, Chap no 13 Page no.383-387 | CO3 | Lab Submission and Class Test-2 and TEE |
| 33 | Representation of graph: Adjacency Matrix, Adjacency List, | Understand | TB2, Chap no 13 Page no.388-393 | CO3 |
| 34 | Graph Traversal: Breadth first search, | APPLY | TB2, Chap no 13 Page no.394-396 | CO3 |
| 35 | Graph Traversal: Depth first search, |  | APPLY | TB2, Chap no 13 Page no.397-400 | CO3 |
| 36 | Applications of Graphs (Problem Solving): Shortest path (Dijkstra’s algorithm) Numerical |  | APPLY | TB2, Chap no 13 Page no.413-414 | CO3 |  |
| 37 | Minimum Spanning Tree (Prim’s and Kruskal) |  | APPLY | TB2, Chap no 13 Page no.405-413 | CO3 |  |
| **Unit 6: Searching and Sorting** | | | | | | |
| 38 | Linear Search | PPT, Whiteboard | ANALYZE | TB2, Chap no 14 Page no.424-425 | CO4 | -  Lab Submissions and TEE |
| 39 | Binary Search | ANALYZE | TB2, Chap no 14 Page no.426-428 | CO4 |
| 40 | Selection Sort |  | ANALYZE | TB2, Chap no 14 Page no.440-442 | CO4 |
| 41 | Insertion sort |  | ANALYZE | TB2, Chap no 14 Page no.438-440 | CO4 |
| 42 | Merge sort |  | ANALYZE | TB2, Chap no 14 Page no.443-446 | CO4 |
| 43 | Introduction to Hashing (Concept and Numerical on Linear, quadratic probing and chaining) |  | ANALYZE | TB2, Chap no 15 Page no.466-479 | CO4 |
| 44 | Solving numerical on Hashing |  | ANALYZE |  | CO4 |  |
| 45 | Solving numerical on Hashing |  | ANALYZE |  | CO4 |  |

1. **Assessment and Evaluation Scheme**

**,**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Internal Continuous Assessment (ICA) (50 Marks)** | | | | |
|  | **Class Tests (20 Marks)** | | **Term-work (30 Marks)** | | |
| **Assessment Component** | Class Test-1 | Class Test-2 | Lab Submissions | Group activity:  projects or Open-problem solving) | Lab exam |
| **Marks** | 10 | 10 | 10 | 10 | 10 |

* 1. **Internal Continuous Assessment (ICA) – 50 marks**

Class Test-1 conducted in the 6th week. It will be for 10 marks (45 min duration). Class Test-2 conducted in the 11th week. It will be for 10 marks (45 min duration). Marks obtained scaled down to 10 for ICA computation. Lab submissions will have a weightage of 10 marks. Evaluation based on timely submissions of programming assignments given every week. Group activity: projects or problem-solving will have weightage of 10 marks. The group activity will be passed on solving an open problem in competitive programming platform similar to CodeChef or doing a mini-project for a given problem statement. Lab exam will be conducted one week before class-test 2 and will have weightage of 10 marks. **50% marks would be deducted for late submissions. (It is important that all assignments and lab submissions are done before the deadline given)**.

* 1. **Term End Examination (TEE) – (100 marks scaled down to 50)**

TEE conducted at the end of the semester will be for 100 marks (3-hour duration). Marks obtained scaled down to 50. There will be 7 questions (each question will be 20 marks). Q1 will be compulsory and any 4 from the remaining to be solved.

* 1. **Course Passing Criteria**
     1. ICA (50 marks) – No minimum marks
     2. TEE (100 marks scaled to 50) – 40% required for passing
     3. (ICA + TEE) (100 marks) - 40% required for passing
  2. **Assessments and Mapping to Course Outcomes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Internal Continuous Assessment (ICA)** | | | | | **Term End Examination (TEE)** |
| Course Outcomes | CT-1 | CT-2 | Lab Submissions | Assignments | Lab Exam | TEE |
| CO-1 | Y |  |  | Y |  | Y |
| CO-2 | Y | Y | Y | Y | Y | Y |
| CO-3 |  | Y | Y | Y | Y | Y |
| CO-4 |  |  | Y | Y |  | Y |

1. **Laboratory details**

The following 10 programming exercises will form the submission for laboratory coursework. The preferred programming language is C++.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exp. No.** | **Week No.#** | **Programming Topic** | **Mapped CO** |
|  | 1 | Implementation of various array operations like traversal, insertion and deletion using any real life application. | CO1 |
|  | 2 | To study and implement concept of Stack data structure and use in recursion | CO2 |
|  | 3 | Application of stack (Expression conversion/evaluation of postfix expression /Parenthesis matching ) | CO2 |
|  | 4 | Implementation of simple queues and circular queue using arrays and apply them in real life computer applications - Railway reservation system | CO2 |
|  | 5 | To study and implement the concept of Linked list data structure. | CO2 |
|  | 6 | Representation of Binary trees in memory and implementation of Binary Tree traversal algorithms | CO3 |
|  | 7 | Implementation of Binary Search Tree: Insertion, deletion and Search operation on tree data structure | CO3 |
|  | 8 | Practical Examination with viva |  |
| 8. | 9 | Implementation of Graph Traversal: Breadth first search, Depth first search | CO3 |
| 9. | 10 | Implementation of Linear search and binary search, understand their difference | CO4 |
| 10 | 11 | Sorting (Any two) |  |

1. **Tutorial Plan**

# This course does not have any tutorial.

1. **Course Material**

References and Lab Manuals would be uploaded on LMS by faculty every week.

1. **GenAI Usage**
   1. **Pre-class Activity: Importance of DSA and its applications**
   2. **In-Class Activity:**
2. Gen AI will not be used to implement code in lab.
3. After writing your own code compare with GenAI generated code and test the Gen AI generated code using various test cases.
   1. **Assignments**

GenAI is not to be used for doing Assignments given by faculty.

1. **Academic Integrity Statement**

Original work expected from students for all of the **assigned assessment work**. Copying in any form not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Plagiarism detection software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment**.**

**\* - Only Teaching Method in the Instruction Plan for the course may vary for different faculty teaching the course, rest of the Course Policy Document will not change.**