| **Course Code** | 18CSC201J | **Course Name** | DATA STRUCTURES AND ALGORITHMS | **Course Category** | *C* | *Professional Core* | L | T | P | C |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 0 | 2 | 4 |

| **Pre-requisite Courses** | *Nil* | | **Co-requisite Courses** | *Nil* | | **Progressive Courses** | *18CSC204J* |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Offering Department** | | *Computer Science and Engineering* | | | **Data Book / Codes/Standards** | *Nil* | |

| **Course Learning Rationale (CLR):** | | | *The purpose of learning this course is to:* | |  | **Learning** | | |  | **Program Learning Outcomes (PLO)** | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |
| **CLR-1 :** | *Utilize the different data types; Utilize searching and sorting algorithms for data search* | | | |  | 1 | 2 | 3 |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **CLR-2 :** | *Utilize linked list in developing applications* | | | |  | Level of Thinking (Bloom) | Expected Proficiency (%) | Expected Attainment (%) |  | Engineering Knowledge | Problem Analysis | Design & Development | Analysis, Design, Research | Modern Tool Usage | Society & Culture | Environment & Sustainability | Ethics | Individual & Team Work | Communication | Project Mgt. & Finance | Life Long Learning | PSO - 1 | PSO - 2 | PSO – 3 |
| **CLR-3 :** | *Utilize stack and queues in processing data for real-time applications* | | | |  |  |
| **CLR-4 :** | *Utilize tree data storage structure for real-time applications* | | | |  |  |
| **CLR-5 :** | *Utilize algorithms to find shortest data search in graphs for real-time application development* | | | |  |  |
| **CLR-6 :** | *Utilize the different types of data structures and its operations for real-time programming applications* | | | |  |  |
|  | |  | | |  |  |
| **Course Learning Outcomes (CLO):** | | | | *At the end of this course, learners will be able to:* | |  |
| **CLO-1 :** | *Identify linear and non-linear data structures. Create algorithms for searching and sorting* | | | | | *3* | *80* | *70* |  | *L* | *H* | *-* | *H* | *L* | *-* | *-* | *-* | *L* | *L* | *-* | *H* | *-* | *-* | *-* |
| **CLO-2 :** | *Create the different types of linked lists and evaluate its operations* | | | | | *3* | *85* | *75* |  | *M* | *H* | *L* | *M* | *L* | *-* | *-* | *-* | *M* | *L* | *-* | *H* | *-* | *-* | *-* |
| **CLO-3 :** | *Construct stack and queue data structures and evaluate its operations* | | | | | *3* | *75* | *70* |  | *M* | *H* | *M* | *H* | *L* | *-* | *-* | *-* | *M* | *L* | *-* | *H* | *-* | *-* | *-* |
| **CLO-4 :** | *Create tree data structures and evaluate its types and operations* | | | | | *3* | *85* | *80* |  | *M* | *H* | *M* | *H* | *L* | *-* | *-* | *-* | *M* | *L* | *-* | *H* | *-* | *-* | *-* |
| **CLO-5 :** | *Create graph data structure, evaluate its operations, implement algorithms to identify shortest path* | | | | | *3* | *85* | *75* |  | *H* | *H* | *M* | *H* | *L* | *-* | *-* | *-* | *M* | *L* | *-* | *H* | *-* | *-* | *-* |
| **CLO-6 :** | *Construct the different data structures and evaluate their types and operations* | | | | | *3* | *80* | *70* |  | *L* | *H* | *-* | *H* | *L* | *-* | *-* | *-* | *L* | *L* | *-* | *H* | *-* | *-* | *-* |

| Duration (hour) | | **15** | **15** | **15** | **15** | **15** |
| --- | --- | --- | --- | --- | --- | --- |
| **S-1** | SLO-1 | *Introduction-Basic Terminology* | *Array* | *Stack ADT* | *General Trees* | *Graph Terminology* |
| SLO-2 | *Data Structures* | *Operations on Arrays – Insertion and Deletion* | *Stack Array Implementation* | *Tree Terminologies* | *Graph Traversal* |
| **S-2** | SLO-1 | *Data Structure Operations* | *Applications on Arrays* | *Stack Linked List Implementation* | *Tree Representation* | *Topological sorting* |
| SLO-2 | *ADT* | *Multidimensional Arrays- Sparse Matrix* | *Applications of Stack- Infix to Postfix Conversion* | *Tree Traversal* | *Minimum spanning tree – Prims Algorithm* |
| **S-3** | SLO-1 | *Algorithms – Searching techniques* | *Linked List Implementation - Insertion* | *Applications of Stack- Postfix Evaluation* | *Binary Tree Representation* | *Minimum Spanning Tree - Kruskal’s Algorithm* |
| SLO-2 | *Complexity – Time , Space Trade off* | *Linked List- Deletion and Search* | *Applications of Stack- Balancing symbols* | *Expression Trees* | *Network flow problem* |
| **S**  **4-5** | SLO-1 | *Lab 1: Implementation of Searching - Linear and Binary Search Techniques* | *Lab 4 : Implementation of Array – Insertion, Deletion.* | *Lab 7 :Implementation of stack using array and Linked List* | *Lab 10: Implementation of Tree using array* | *Lab 13: Implementation of Graph using Array* |
| SLO-2 |
| **S-6** | SLO-1 | *Algorithms - Sorting* | *Applications of Linked List* | *Applications of Stack- Nested Function Calls* | *Binary Tree Traversal* | *Shortest Path Algorithm- Introduction* |
| SLO-2 | *Complexity – Time , Space Trade off* | *Polynomial Arithmetic* | *Recursion concept using stack* | *Threaded Binary Tree* | *Shortest Path Algorithm: Dijkstra’s Algorithm* |
| **S-7** | SLO-1 | *Mathematical notations* | *Cursor Based Implementation – Methodology* | *Applications of Recursion: Tower of Hanoi* | *Binary Search Tree :Construction, Searching* | *Hashing: Hash functions - Introduction* |
| SLO-2 | *Asymptotic notations-Big O, Omega* | *Cursor Based Implementation* | *Queue ADT* | *Binary Search Tree : Insertion and Deletion* | *Hashing: Hash functions* |
| **S-8** | SLO-1 | *Asymptotic notations - Theta* | *Circular Linked List* | *Queue Implementation using array* | *AVL Trees: Rotations* | *Hashing : Collision avoidance* |
| SLO-2 | *Mathematical functions* | *Circular Linked List - Implementation* | *Queue Implementation using Linked List* | *AVL Tree: Insertions* | *Hashing : Separate chaining* |
| **S**  **9-10** | SLO-1 | *Lab 2: Implementation of sorting Techniques – Insertion sort and Bubble Sort Techniques* | *Lab 5: Implementation of Linked List - Cursor Based Implementation* | *Lab 8: Implementation of Queue using Array and linked list* | *Lab 11:*  *Implementation of BST using linked list* | *Lab 14 :Implementation of Shortest path Algorithm* |
| SLO-2 |
| **S-11** | SLO-1 | *Data Structures and its Types* | *Applications of Circular List -Joseph Problem* | *Circular Queue* | *B-Trees Constructions* | *Open Addressing* |
| SLO-2 | *Linear and Non-Linear Data Structures* | *Doubly Linked List* | *Implementation of Circular Queue* | *B-Trees Search* | *Linear Probing* |
| **S-12** | SLO-1 | *1D, 2D Array Initialization using Pointers* | *Doubly Linked List Insertion* | *Applications of Queue* | *B-Trees Deletions* | *Quadratic probing* |
| SLO-2 | *1D, 2D Array Accessing using Pointers* | *Doubly Linked List Insertion variations* | *Double ended queue* | *Splay Trees* | *Double Hashing* |
| **S-13** | SLO-1 | *Declaring Structure and accessing* | *Doubly Linked List Deletion* | *Priority Queue* | *Red Black Trees* | *Rehashing* |
| SLO-2 | *Declaring Arrays of Structures and accessing* | *Doubly Linked List Search* | *Priority Queue - Applications* | *Red Black Trees Insertion* | *Extensible Hashing* |
| **S**  **14-15** | SLO-1 | *Lab 3: Implement Structures using Pointers* | *Lab 6: Implementation of Doubly linked List* | *Lab 9: Applications of Stack, Queue* | *Lab 12:Implementation of B-Trees* | *Lab 15 :Implementation of Minimal Spanning Tree* |
| SLO-2 |

| **Learning**  **Resources** | 1. *Seymour Lipschutz, Data Structures with C, McGraw Hill, 2014* 2. *R.F.Gilberg, B.A.Forouzan, Data Structures, 2nd ed., Thomson India, 2005* 3. *A.V.Aho, J.E Hopcroft , J.D.Ullman, Data structures and Algorithms, Pearson Education, 2003* 4. *Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd ed., Pearson Education, 2015* | 1. *Reema Thareja, Data Structures Using C, 1st ed., Oxford Higher Education, 2011* 2. *Thomas H Cormen, Charles E Leiserson, Ronald L Revest, Clifford Stein, Introduction to Algorithms 3rd ed., The MIT Press Cambridge, 2014* |
| --- | --- | --- |

| **Learning Assessment** | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bloom’s  Level of Thinking | Continuous Learning Assessment (50% weightage) | | | | | | | | Final Examination (50% weightage) | |
| CLA – 1 (10%) | | CLA – 2 (15%) | | CLA – 3 (15%) | | CLA – 4 (10%)# | |
| Theory | Practice | Theory | Practice | Theory | Practice | Theory | Practice | Theory | Practice |
| Level 1 | Remember | *20%* | *20%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* |
| Understand |
| Level 2 | Apply | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* | *20%* |
| Analyze |
| Level 3 | Evaluate | *10%* | *10%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* | *15%* |
| Create |
|  | Total | 100 % | | 100 % | | 100 % | | 100 % | | - | |

# CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

| **Course Designers** |  |  |
| --- | --- | --- |
| Experts from Industry | Experts from Higher Technical Institutions | Internal Experts |
| *1. Dr. Nagaveer, CEO, Campus Corporate Connect,* *nagaveer@campuscorporateconnect.com* | *1. Dr. Srinivasa Rao Bakshi, IITM, Chennai, sbakshi@iitm.ac.in* | *1. Mr. K. Venkatesh, SRMIST* |
| *2. Dr. Sricharan Srinivasan, Wipro Technologies, sricharanms@gmail.com* | *2. Dr. Ramesh Babu, N , nrbabu@iitm.ac.in* | *2. Dr.Subalalitha C.N, SRMIST* |
|  | *3. Dr.Noor Mahammad, IIITDM, Kancheepuram,* *noor@iiitdm.ac.in* | *3. Ms. Ferni Ukrit, SRMIST* |