|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Code | | | 21CSC201J | | | **Course Name** | | | | DATA STRUCTURES AND ALGORITHMS | | **Course Category** | | | *C* | | | *Professional Core* | | | | | | | | | | | | L | | T | | P | | C | | |
| 3 | | 0 | | 2 | | 4 | | |
| **Pre-requisite Courses** | | | | *Principles of Programming Practices* | | | | | **Co-requisite Courses** | | *Nil* | | | | | **Progressive Courses** | | | | | | *Design and Analysis of Algorithms* | | | | | | | | | | | | | | | |
| **Course Offering Department** | | | | | | | | *Department of Computing Technologies* | | | **Data Book / Codes/Standards** | | | | | *Nil* | | | | | | | | | | | | | | | | | | | | | |
| **Course Learning Rationale (CLR):** | | | | | | | *The purpose of learning this course is to:* | | | | | |  |  | | | **Program Outcomes (PO)** | | | | | | | | | | | | | | | | | | | | |
|  |  | | |
| **CLR-1 :** | Structures, pointers, searching, sorting techniques used to handle a set of data along with time and space complexity | | | | | | | | | | | |  |  | | | 1 | | 2 | 3 | 4 | | 5 | 6 | 7 | 8 | 9 | 10 | 11 | | 12 | | 13 | | 14 | | 15 |
| **CLR-2 :** | List structure and its categories | | | | | | | | | | | |  |  | | | 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 :** | Linear structures Stack and Queue | | | | | | | | | | | |  |  | | |
| **CLR-4 :** | Tree structure with its applications and hashing methods | | | | | | | | | | | |  |  | | |
| **CLR-5 :** | Structures Graphs and implement them | | | | | | | | | | | |  |  | | |
| **CLR-6 :** | Different types of data structures and use them for problem solving | | | | | | | | | | | |  |  | | |
|  | |  | | | | | | | | | | |  |  | | |
| **Course Outcomes (CO):** | | | | | *At the end of this course, learners will be able to:* | | | | | | | | |  | | |
| **CO-1 :** | Develop programs using data types like structures, pointers and arrays supported by C programming language | | | | | | | | | | | | |  | | | *1* | | *-* | *3* | *-* | | *-* | *-* | *-* | *-* | *2* | *-* | *-* | | *-* | | *1* | | *-* | | *2* |
| **CO-2 :** | Analyze the complexity of algorithm and if needed, modify it to improve its efficiency | | | | | | | | | | | | |  | | | *2* | | *3* | *-* | *1* | | *-* | *-* | *-* | *-* | *-* | *-* | *-* | | *-* | | *1* | | *2* | | *-* |
| **CO-3 :** | Identify and Use appropriate data structure for devising solution | | | | | | | | | | | | |  | | | *1* | | *3* | *2* | *-* | | *-* | *-* | *-* | *-* | *-* | *-* | *-* | | *-* | | *1* | | *1* | | *2* |
| **CO-4 :** | Describe and use tree structure while developing programs | | | | | | | | | | | | |  | | | *2* | | *-* | *3* | *2* | | *-* | *-* | *-* | *-* | *-* | *-* | *-* | | *-* | | *1* | | *-* | | *2* |
| **CO-5 :** | Implement the Graph structure and use it whenever deemed necessary for providing better solution | | | | | | | | | | | | |  | | | *3* | | *2* | *3* | *-* | | *-* | *-* | *-* | *-* | *-* | *-* | *-* | | *-* | | *1* | | *1* | | *2* |
| **CO-6 :** | Decide and use appropriate searching and sorting algorithms while developing solutions for specific problems | | | | | | | | | | | | |  | | | *1* | | *3* | *-* | *2* | | *-* | *-* | *-* | *-* | *-* | *-* | *-* | | *-* | | *1* | | *1* | | *1* |

|  |
| --- |
| Unit-1 Introduction  Programming in C - Primitive data types, Structures, Self-referential structures, Pointers and structures, Dynamic memory allocation, Matrix multiplication; Data Structure – Definition, Types, ADT, Operations; Mathematical notations - Big O, Omega and Theta,  Complexity – Time, Space, Trade off. |
| Unit-2 List Structure  Operations on List ADT – Create, Insert, Search, Delete, Display elements; Implementation of List ADT– Array, Cursor based and Linked; Types – Singly, Doubly, Circular; Applications - Sparse Matrix, Polynomial Arithmetic, Joseph Problem |
| Unit-3 Stack and Queue  Operations on Stack ADT – Create, Push, Pop, Top; Implementation of Stack ADT – Array and Linked; Applications - Infix to Postfix Conversion, Postfix Evaluation, Balancing symbols, Function Calls, Tower of Hanoi; Operations on Queue ADT - Create, Enqueue and Dequeue; Implementation of Queue ADT – Array and Linked; Types of Queue - Circular, Double ended and Priority Queue, Applications – Scheduling |
| Unit-4 Trees and Hashing  Introduction to Trees, Tree traversals, Complete Binary Tree and its height, Binary Search Trees, Need for Balance, Rotation, AVL trees, B Trees, Heaps, trees and array implementations and applications; Hash functions - Introduction, functions, Collision  avoidance, Separate chaining, Open Addressing, Linear Probing, Quadratic probing. |
| Unit-5 Graph  Introduction to Graph, Graph Traversal, Topological sorting, Minimum spanning tree – Prims Algorithm, Kruskal’s Algorithm, Shortest Path Algorithm - Dijkstra’s Algorithm |
| Lab  Lab 1: Implementation of Structures  Lab 2: Implementation of Structures using Pointers  Lab 3: Implementation of Matrix Multiplication – Dynamic Memory allocation  Lab 4: Array Implementation of List  Lab 5: Implementation of Linked List  Lab 6: Implementation of Doubly linked List  Lab 7: Implementation of Stack using array and Linked List  Lab 8: Implementation of Queue using array and Linked list  Lab 9: Applications of Stack, Queue  Lab 10: Implementation of Tree using array  Lab 11: Implementation of BST using linked list  Lab 12: Implementation of B-Trees  Lab 13: Implementation of Graph using Array  Lab 14: Implementation of Shortest path Algorithm  Lab 15: Implementation of Minimal Spanning Tree |

|  |  |  |
| --- | --- | --- |
| 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 | 1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd ed., Pearson Education, 2015 2. Reema Thareja, Data Structures Using C, 1st ed., Oxford Higher Education, 2011 3. Thomas H Cormen, Charles E Leiserson, Ronald L Revest, Clifford Stein, Introduction to Algorithms 3rd ed., The MIT Press Cambridge, 2014 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
|  | Bloom’s  Level of Thinking | **Continuous Learning Assessment (CLA)**  **- By the Course Faculty** | | | | **By The CoE** | |
| **Formative**  CLA-I Average of  unit test  (50%) | | **Life Long\***  **Learning**  CLA-II- Practice  (10%) | | **Summative**  Final  Examination  (40% weightage) | |
| Theory | Practice | Theory | Practice | Theory | Practice |
| Level 1 | Remember | *25%* | - | - | *10%* | *25%* | - |
| Level 2 | Understand | *25%* | - | - | *20%* | *25%* | - |
| Level 3 | Apply | *20%* | - | - | *30%* | *20%* | - |
| Level 4 | Analyze | *20%* | - | - | *30%* | *20%* | - |
| Level 5 | Evaluate | *10%* | - | - | *10%* | *10%* | - |
| Level 6 | Create | - | - | - | - | - | - |
|  | Total | 100 % | | 100 % | | 100 % | |

|  |  |  |
| --- | --- | --- |
| Course Designers |  |  |
| Experts from Industry | Experts from Higher Technical Institutions | Internal Experts |
| Dr. Mariappan Vaithilingam, Senior Engineering Manager, Uber India Research and Development Pvt Centre, Bangalore. | Dr. Venkatesh Raman, Professor  Mathematical Institute of Science | 1. Dr. K. Vijaya, SRMIST |
|  |  | 2. Dr. S. Poornima, SRMIST |
|  |  | 3. Dr. K. Venkatesh, SRMIST |