Course Code	21CSC201J	Course Name	Data Structures and Algorithms	Course Category		Professional Core	L	T	P	C
							3	0	2	4

Pre-requisite Courses	Nil	Co-requisite Courses	Nil	Progressive Courses	Nil
Course Offering Department	Computing Technologies	*Data Book / Codes / Standards*	Nil		

Course Learning Rationale (CLR):	*The purpose of learning this course is to:*
CLR-1 :	Know about searching and sorting techniques used to handle a set of data along with time and space complexity
CLR-2 :	Utilize various categories of list structures to develop solutions
CLR-3 :	Explore usage of Stack and Queues in processing data for real time applications
CLR-4 :	Understand tree structure and its applications
CLR-5 :	Utilize hash tables for data storage and use graphs to solve real time problems

Course Outcomes (CO):	*At the end of this course, learners will be able to:*	Program Outcomes (PO)												Program Specific outcomes		
		1	2	3	4	5	6	7	8	9	10	11	12	PSO-1	PSO-2	PSO-3
		Engineering Knowledge	Problem Analysis	Design/development of solutions	Conduct investigations of complex problems	Modern Tool Usage	The engineer and society	Environment & Sustainability	Ethics	Individual & Team Work	Communication	Project Mgt. & Finance	Life Long Learning			
CO-1:	Devise algorithms to arrange the data in required order and retrieve a specific datum in efficient manner	1	2	3	-	-	-	-	-	-	-	-	-	3	-	-

CO-2:	Determine the type of list structure that could be used for solving a problem and implement it using C programming language	2	3	3	-	-	-	-	-	-	-	-	--	3	-	-
CO-3:	Devise solutions using linear structures Stack and Queue	2	3	3	-	-	-	-	-	-	-	-	-	3	-	-
CO-4:	Express proficiency in usage of tree for solving problems	2	3	3	-	-	-	-	-	-	-	-	-	3	-	-
CO-5:	Implement Hash tables for storing data and algorithms to find shortest path between nodes in a graph	3	2'	3	-	-	-	-	-	-	-	-	-	3	-	-

Unit-1 : Introduction

Primitive data types, Structures, Self-referential structures, Pointers, Dynamic memory allocation, Matrix multiplication; Data Structure – Definition, Types, ADT, Operations; Mathematical notations, Asymptotic Notations-Big O, Omega and Theta; Complexity – Time, Space, Trade off, Searching-Linear Search and Binary Search; Sorting-Insertion sort, Bubble sort.

Unit-2 : Array and Linked List

Operations on List ADT – Create, Insert, Search, Delete, Display elements; Implementation of List ADT– Array, Cursor based Linked List; Types – Singly Linked List, Doubly Linked List, Circular Linked List; Applications - Sparse Matrix, Polynomial Arithmetic, Josephus Problem.

Unit-3: Stack and Queues

Operations on Stack ADT – Create, Push, Pop, Top; Implementation of Stack ADT – Array and Linked List; 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 List; Types of Queue - Circular, Double ended and Priority Queue, Applications – Scheduling.

Unit-4 : Trees

Introduction to Trees, Tree traversals, Binary Tree Representation, Complete Binary Tree, Binary Search Trees-Searching, Insertion and Deletion, Self-Balancing BST - AVL trees, B trees, Red-black tree, Heap.

Unit-5: Hashing and Graphs

Hashing - Introduction, Hash functions, Collision avoidance, Separate chaining, Open Addressing, Linear Probing, Quadratic probing, Introduction to Graph, Graph Traversal, Topological sorting, Minimum spanning tree – Prim's Algorithm, Kruskal's Algorithm, Shortest Path Algorithm - Dijkstra's Algorithm.

Lab:
Lab 1: Problem solving using structures and pointers
Lab 2: Implementation of sorting and searching
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 - Shortest path Algorithm Lab 14: Implementation of Minimal Spanning Tree Lab 15: Implementation of hash tables

Learning Resources	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, 201	4. A.V. Aho, J.E Hopcroft , J.D.Ullman, Data structures and Algorithms, Pearson Education, 2003 5. Seymour Lipschutz, Data Structures with C, McGraw Hill, 2014 6. R.F.Gilberg, B.A.Forouzan, Data Structures, 2nd ed., Thomson India, 2005

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (CLA)				Summative Final Examination (40% weightage)	
		Formative CLA-I Average of unit test (45%)		Life Long* Learning CLA-II- Practice (15%)			
		Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	15%	-	-	10%	25%	-
Level 2	Understand	30%	-	-	25%	30%	-
Level 3	Apply	30%	-	-	30%	25%	-
Level 4	Analyze	25%	-	-	25%	20%	-
Level 5	Evaluate	-	-	-	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, Theoretical Computer Science, Institute of Mathematical Sciences	1. Dr. K. Vijaya, Associate Professor, SRMIST 2. Dr. S. Poornima, Assistant Professor, SRMIST 3. Dr. P. Saranya, Assistant Professor, SRMIST 4. Dr. K. Venkatesh, Assistant Professor, SRMIST