Course Code	21CSC204J	Course Name	DESIGN AND ANALYSIS OF ALGORITHMS	Course Category	C	PROFESSIONAL CORE	L	T	P	C
							3	0	2	4

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

Course Learning Rationale (CLR):	***The purpose of learning this course is to:***
CLR-1:	*design efficient algorithms in solving complex real time problems*
CLR-2:	*analyze various algorithm design techniques to solve real time problems in polynomial time*
CLR-3:	*utilize various approaches to solve greedy and dynamic algorithms*
CLR-4:	*utilize back tracking and branch and bound paradigms to solve exponential time problems*
CLR-5:	*analyze the need of approximation and randomization algorithms, utilize the importance Non polynomial algorithms*

		Program Outcomes (PO)												**Program Specific Outcomes**		
		1	2	3	4	5	6	7	8	9	10	11	12			
Course Outcomes (CO):	***At the end of this course, learners will be able to:***	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	PSO-1	PSO-2	PSO-3
CO-1:	*apply efficient algorithms to reduce space and time complexity of both recurrent and non-recurrent relations*	2	1	2	1	-	-	-	-	-	3	-	3	3	1	-
CO-2:	*solve problems using divide and conquer approaches*	2	1	2	1	-	-	-	-	-	3	-	3	3	1	-
CO-3:	*apply greedy and dynamic programming type's techniques to solve polynomial time problems*	2	1	2	1	-	-	-	-	-	3	-	3	3	1	-
CO-4:	*create exponential problems using backtracking and branch and bound approaches*	2	1	2	1	-	-	-	-	-	3	-	3	3	1	-
CO-5:	*interpret various approximation algorithms and interpret solutions to evaluate P type, NP Type, NPC, NP Hard problems*	2	1	2	1	-	-	-	-	-	3	-	3	3	1	-

Unit-1 - Introduction to Algorithm Design ***15 Hour***

Fundamentals of Algorithms- Correctness of algorithm - Time complexity analysis - Insertion sort-Line count, Operation count Algorithm Design paradigms - Designing an algorithm And its analysis-Best, Worst and Average case - Asymptotic notations Based on growth functions. O,O,Θ, ω, Ω - Mathematical analysis - Induction, Recurrence relations -Solution of recurrence relations - Substitution method - Solution of recurrence relations - Recursion tree - Solution of recurrence relations - examples.

Unit-2 - Divide and Conquer ***15 Hour***

Maximum Subarray Problem Binary Search - Complexity of binary search Merge sort - Time complexity analysis -Quick sort and its Time complexity analysis Best case, Worst case, Average case analysis - Strassen's Matrix multiplication and its recurrence relation - Time complexity analysis of Merge sort - Largest sub-array sum - Time complexity analysis of Largest sub- array sum - Master Theorem Proof - Master theorem examples - Finding Maximum and Minimum in an array - Time complexity analysis-Examples - Algorithm for finding closest pair problem - Convex Hull problem

Unit-3 - Greedy and Dynamic Programming ***15 Hour***

- Examples of problems that can be solved by using greedy and dynamic approach Huffman coding using greedy approach Comparison of brute force and Huffman method of encoding - Knapsack problem using greedy approach Complexity derivation of knapsack using greedy - Tree traversals - Minimum spanning tree – greedy Kruskal's algorithm - greedy - Minimum spanning tree - Prims algorithm Introduction to dynamic programming - 0/1 knapsack problem - Complexity calculation of knapsack problem - Matrix chain multiplication using dynamic programming - Complexity of matrix chain multiplication - Longest common subsequence using dynamic programming - Explanation of LCS with an example - Optimal binary search tree (OBST)using dynamic programming - Explanation of OBST with an example.

Unit-4 - Backtracking	**15 Hour**
branch and bound - N queen's problem – backtracking - Sum of subsets using backtracking Complexity calculation of sum of subsets Graph introduction Hamiltonian circuit - backtracking - Branch and bound - Knapsack problem Example and complexity calculation. Differentiate with dynamic and greedy Travelling salesman problem using branch and bound - Travelling salesman problem using branch and bound example - Travelling salesman problem using branch and bound example - Time complexity calculation with an example - Graph algorithms - Depth first search and Breadth first search - Shortest path introduction - Floyd-Warshall Introduction - Floyd-Warshall with sample graph - Floyd-Warshall complexity	
Unit-5 - Randomized and Approximation Algorithm	**15 Hour**
Randomized hiring problem Randomized quick sort Complexity analysis String matching algorithm Examples - Rabin Karp algorithm for string matching Example discussion - Approximation algorithm - Vertex covering - Introduction Complexity classes - P type problems - Introduction to NP type problems - Hamiltonian cycle problem - NP complete problem introduction - Satisfiability problem - NP hard problems – Examples	

Lab Experiments	
Lab 1: Simple Algorithm-Insertion sort	*Lab 9: Longest common subsequence*
Lab 2: Bubble Sort	*Lab 10: N queen's problem*
Lab 3: Recurrence Type-Merge sort, Linear search	*Lab 11: Travelling salesman problem*
Lab 4: Quicksort, Binary search	*Lab 12: BFS and DFS implementation with array*
Lab 5: Strassen Matrix multiplication	*Lab 13: Randomized quick sort*
Lab 6: Finding Maximum and Minimum in an array, Convex Hull problem	*Lab 14: String matching algorithms*
Lab 7: Huffman coding, knapsack and using greedy	*Lab 15: Discussion over analyzing a real time problem*
Lab 8: Various tree traversals,	

Learning Resources	1. *Thomas H Cormen, Charles E Leiserson, Ronald L Revest, Clifford Stein, Introduction to Algorithms, 3rd ed., The MIT Press Cambridge, 2014* 2. *Mark Allen Weiss, Data Structures and Algorithm Analysis in C, 2nd ed., Pearson Education, 2006*	3. *Ellis Horowitz, Sartajsahni, Sanguthevar, Rajesekaran, Fundamentals of Computer Algorithms, Galgotia Publication, 2010* 4. *S. Sridhar, Design and Analysis of Algorithms, Oxford University Press, 2015*

Learning Assessment

	Bloom's Level of Thinking	Continuous Learning Assessment (CLA)				Summative Final Examination (40% weightage)	
		Formative CLA-1 Average of unit test (45%)		Life-Long Learning CLA-2 (15%)			
		Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	30%	-	-	30%	30%	-
Level 2	Understand	70%	-	-	30%	30%	-
Level 3	Apply	-	-	-	40%	40%	-
Level 4	Analyze	-	-	-	-	-	-
Level 5	Evaluate	-	-	-	-	-	-
Level 6	Create	-	-	-	-	-	-
	Total	100 %		100 %		100 %	

Course Designers

Experts from Industry	**Experts from Higher Technical Institutions**	**Internal Experts**
1. *G. Venkiteswaran, Wipro Technologies, gvenki@pilani.bits-pilani.ac.in*	1. *Mitesh Khapra, IITM Chennai, miteshk@cse.iitm.ac.in*	1. *Dr. K.Senthil Kumar, SRMIST*
2. *Dr. Sainarayanan Gopalakrishnan, HCL Technologies, sai.jgk@gmail.com*	2. *V. Masilamani. IIITDM, masila@iiitdm.ac.in*	2. *Dr. V. Sivakumar, SRMIST*
		3. *Dr. R. Vidhya,SRMIST*