Detailed Syllabus
Lab Session-wise Breakup

Subject Code	15B17CI471	**Semester Even (specify ~~Odd~~/Even)**	**Semester** IV **Session** 2021-2022 **Month from:** Jan **to** June 2022
Subject Name	Algorithms and Problem Solving Lab		
Credits	**1**	**Contact Hours**	**2**

Faculty (Names)	**Coordinator(s)**	Sherry Garg(62), Akanksha Mehndiratta (128)
	Teacher(s) (Alphabetically)	**J62:** Ankita Wadhwa, Bharat Gupta, Dhanalekshmi G, Hema N, Mahendra Gurve, Nitish A, Jyoti, Tribhubhan K Tiwari, Sherry Garg, Suma Dawn, Vivek K Singh **J128:** Akanksha Mehndiratta, Himani Bansal, Pulkit Mehndiratta, Raju Pal, Shikha Mehta, Surendra Kumar

COURSE OUTCOMES		**COGNITIVE LEVELS**
C274.1	Choose and define appropriate data structure to a given problem	Remember Level (Level 1)
C274.2	Understand various data structures and algorithm design techniques with the help of examples.	Understand Level (Level 2)
C274.3	Apply and build various algorithms and design techniques to solve the given problem.	Apply Level (Level 3)
C274.4	Analyze the algorithm by their complexity using asymptotic analysis.	Analyze Level (Level 4)
C274.5	Evaluate the correctness and complexity of the algorithm for a given problem.	Analyze Level (Level 4)
C274.6	Formulate, elaborate and design an efficient solution to a given problem using appropriate data structure and algorithm design technique	Apply Level (Level 3)

Module No.	**Title of the Module**	**List of Experiments**	**CO**
1.	Analysis of algorithms, Searching and sorting based problems	Introduction to problem solving approach; Asymptotic Analysis; Solving Recurrences; Empirical analysis of sorting and searching algorithms – Merge sort, Quick sort, Heap sort, Radix sort, Count sort, Binary search, and Median search	CO1, CO2, CO3, CO4
2.	Design Technique: Divide and Conquer	Problems based on Divide and Conquer (D&C) approach such as Binary search, Quick sort, and Merge sort; and Closest pair, etc.	CO3, CO5
3.	Design Technique: Greedy Algorithms	Introduction to greedy based solution approach; Minimum Spanning Trees (Prim's and Kruskal algorithms); Shortest Path using Dijkstra's algorithm; Fractional and 0/1 Knapsack; Coinage problem; Bin packing; Job scheduling – Shortest job first, Shortest remaining job first, etc.; Graph coloring; and Text compression using Hamming coding and Shannon-Fano coding, etc.	CO3, CO5
4.	Design Technique: Backtracking Algorithms	Review of backtracking based solution approach using N queen, and Rat in a maze; M-coloring problem; Hamiltonian Cycle detection; Travelling salesman problem; Network flow	CO3, CO5
5.	Dynamic Programming	Fundamentals of Dynamic programming based solution approach; 0/1 Knapsack ; Shortest path using Floyd Warshall; Coinage problem; Matrix Chain Multiplication; Longest	CO3, CO5

		common subsequence; Longest increasing sequence, String editing	
6.	String Algorithms	Naïve String Matching, Finite Automata Matcher, Rabin Karp matching algorithm, Knuth Morris Pratt, Tries; Suffix Tree; and Suffix Array	CO3, CO5
7.	Problem Spaces and Problem solving by search	Problem Spaces: States, goals and operators, Factored representation (factoring state into variables) Uninformed search (BFS, DFS, DFS with iterative deepening), Heuristics and informed search (hill-climbing, generic best-first, A*)	CO3, CO5
8.	Case-study / Assignment / Mini-Project	Designing an efficient solution to a given problem using appropriate data structure and algorithm design technique	CO5, CO6

Evaluation Criteria

Components	**Maximum Marks**
Lab Test 1	**20**
Lab Test 2	**20**
Evaluation 1	**10**
Evaluation 2	**15**
PBL/Mini Project	**20**
Attendance	**15**
Total	**100**

Project based learning: Students in a group of 4-5 will be designing an efficient solution to a given problem / case-studies using appropriate data structure and algorithm design technique studies in the course. The students have to implement the mini project using C/C++/Java language. Project development and its presentation will enhance coding skills, knowledge and employability of the students in IT sector.

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Reference Books,** Journals, Reports, Websites etc. in the IEEE format)

1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 3rd Edition, 2009
2.	Steven Skiena ,The Algorithm Design Manual, Springer; 2nd edition , 2008
3.	Knuth, The art of Computer Programming Volume 1, Fundamental Algorithms, Addison-Wesley Professional; 3 edition,1997
4.	Horowitz and Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 2008
5.	Sedgewick, Algorithms in C, 3rd edition. Addison Wesley, 2002
6.	Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1983
7.	ACM Transactions on Algorithms (TALG)
8.	Algorithmica Journal, Springer
9.	Graphs and Combinatorics, Journal, Springer
10.	The ACM Journal of Experimental Algorithmics

Recommended Reading material: Author(s), Title, Edition, Publisher, Year of Publication etc. **(Text books**)	
1.	Tim Roughgarden, Algorithms Illuminated: Part 1: The Basics, Soundlikeyourself Publishing, September 27, 2017
2.	Tim Roughgarden, Algorithms Illuminated:Part 2: Graph Algorithms and DataStructures ,Soundlikeyourself Publishing, First Edition, 2018.
3.	Tim Roughgarden, Algorithms Illuminated :Part3:Greedy Algorithms and Dynamic Programming,Soundlikeyourself Publishing, First Edition, 2019.
4.	Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition**,** Pearson, 2014