	ITER, SIKSHA 'O' ANUSANDHAN **(Deemed to be University)**		**LESSON PLAN**
Programme	**B.Tech.**	Academic Year	**2023-24**
Department	**CSE**	Semester	**4th**
Credit	**4**	Grading Pattern	**1**
Subject Code	**CSE 4131**		
Subject Name	**ALGORITHM DESIGN-2**		
Weekly Course Format	**3L - 2P**		
Instructor	**Dr. Satya Ranjan Das**		

References Books(s):

(T1) Algorithm Design by Jon Kleinberg and Eva Tardos, Pearson Publication.

(T2) Problem Soving in Data Structures and Algorithms using Java by Hemant Jain.

(T3) The Algorithm Design Manual by Steven S. Skiena, Springer Publication

Course Outcomes	Students will be able to	
	CO1	To understand the Network flow problem and apply it to solve different real-world problems.
	CO2	To distinguish between computationally tractable and intractable problems,define and relate classP, classNP and class NPcomplete,define an appropriate certificate and the verification algorithm for a given problem in NP. .
	CO3	To understand approximation algorithms and apply this concept to solve some problems for which polynomial time exact solutions are probably unattainable.
	CO4	To understand local search techniques and apply this concept to design heuristics for some computationally hard problems.
	CO5	To understand randomization and apply this concept to solve problems.
	CO6	To identify and apply an appropriate algorithmic approach to solve a problem and explain the challenges to solve it.

Sl.No.	Lessons/Topics to be covered	Book Reference (sections)	Mapping with COs	Home Work/ Assignments/ Quizzes
1	Introduction to the course/subject: Lesson plan; Course Goal; Teaching methodology; Evaluation strategy etc.			
2	Review on Time and Space Complexity, Asymptotic Notations, Recurrences, Sorting and Searching			
3	Review on Graph representation, graph traversal, spanning tree and shortest path			
4	**Lab#1:** Java implementation of Graph related algorithms (BFS, DFS, MST, Shortest Path)	T2-12 (pg.405)	**CO1-CO6**	
5	Review on Greedy, DAC and DP Algorithms			**Quiz-1**
6	Weighted graph algorithms: Network flows and Bipartite matching (including all theorems and lemma – 7.1 to 7.5)	T1_7.1-7.2 (pg.338-345)	**CO1**	
7	Weighted graph algorithms: Network flows and Bipartite matching (including all theorems and lemma – 7.6 to 7.14)	T1_7.3 (pg.345-352)	**CO1**	
8	**Lab#2:** Java implementation of Graph related algorithms (BFS, DFS, MST, Shortest Path)	T2-12 (pg.405)	**CO1-CO6**	
9	Weighted graph algorithms: Network flows and Bipartite matching (including all theorems and lemma – 7.34 to 7.38)	T1_7.5 (pg.368-370)	**CO1**	Assignment-1
10	NP and Computational Intractability: Problem Reductions (all subsections)	T3_9.2 (pg. 319-323)	**CO2**	
11	NP and Computational Intractability: Problem Reductions (all subsections)	T3_9.3 (pg. 324-327)	**CO2**	**Quiz-2**
12	**Lab#3:** Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)	T2-13 (pg.454)	**CO1-CO6**	

Sl.No.	Lessons/Topics to be covered	Book Reference (sections)	Mapping with COs	Home Work/ Assignments/ Quizzes
13	NP and Computational Intractability: Problem Reductions (all subsections)	T1_8.1 (pg.452-458)	**CO2**	
14	NP and Computational Intractability: Satisfiability (Formula-SAT, CKT-SAT etc.) (including theorems and lemma – 8.8, 8.9)	T1_802 (pg.460-462)	**CO2**	
15	NP and Computational Intractability: P, NP, NPC and NPH (including theorems and lemma – 8.10, 8.11, 8.12, 8.16)	T1_8.3-8.4 (pg.463-472)	**CO2**	
16	**Lab#4:** Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)	T2-13 (pg.454)	**CO1**	
17	NP and Computational Intractability: P, NP, NPC and NPH	T1_8 (pg.460-472)	**CO2**	
18	PSPACE: A class Problems Beyond NP (including all theorems and lemma – 9.1 to 9.3)	T1_9.1 (pg.531-532)	**CO1**	
19	PSPACE: A class Problems Beyond NP	T_9.2 (pg.533-535)	**CO2**	
20	**Lab#5:** Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)	T2-13 (pg.454)	**CO1-CO6**	
21	PSPACE: A class Problems Beyond NP	T_9.2 (pg.533-535)	**CO2**	
22	Extending the Limits of Intractability (including theorems and lemma – 10.1 to 10.2 – statements only with example, no proof required)	T1_10.1 (pg.553-554)	**CO2**	**Assignment-2**

Sl.No.	Lessons/Topics to be covered	Book Reference (sections)	Mapping with COs	Home Work/ Assignments/ Quizzes
23	Extending the Limits of Intractability (including theorems and lemma – 10.3 to 10.4 – statements only with example, no proof required)	T1_10.1 (pg.555-556)	**CO2**	**Quiz-2**
24	**Lab#6:** Java implementation of Greedy Algorithm (Interval Scheduling, Fractional Knapsack, Huffman Coding)	T2_16 (pg.488)	**CO1-CO6**	
25	Approximation Algorithms (including all theorems and lemma – 11.1 to 11.5)	T1_11.1 (pg.600-605)	**CO3**	
26	Approximation Algorithms (including all theorems and lemma – 11.6 to 11.8)	T1_11.2 (pg.608-611)	**CO3**	
27	Approximation Algorithms (including only the statements of all theorems and lemma – 11.9 to 11.10—no proofs required)	T1_11.3 (pg.612-615)	**CO3**	
28	**Lab#7:** Java implementation of Greedy Algorithm (Interval Scheduling, Fractional Knapsack, Huffman Coding)	T2_16 (pg.454)	**CO1-CO6**	
29	Approximation Algorithms (including all theorems and lemma – 11.12 to 11.13)	T1_11.4 (pg.618-620)	**CO3**	
30	Approximation Algorithms (including all theorems and lemma – 11.14 to 11.15)	T1_11.4 (pg.621-623)	**CO3**	Assignment-3
31	Local Search	T1_12.1 (pg.749)	**CO4**	
32	**Lab#8:** Java implementation of Divide-and-Conquer based algorithms (Merge sort, Counting Inversions, Quick sort)	T2_17 (pg.497)	**CO1-CO6**	
33	Local Search (The landscape of an optimization problem – statement 12.1 with example of vertex cover problem	T1_12.1 (pg.750-753)	**CO4**	

Sl.No.	Lessons/Topics to be covered	Book Reference (sections)	Mapping with COs	Home Work/ Assignments/ Quizzes
34	Local Search (Application of local search to Hopfield Neural Networks – algorithm with example – no proof required)	T1_12.3 (pg.760-761)	**CO4**	
35	Local Search (Application of local search to Hopfield Neural Networks – algorithm with example – no proof required)	T1_12.3 (pg.762-763)	**CO4**	
36	**Lab#9:** Java implementation of Divide-and-Conquer based algorithms (Merge sort, Counting Inversions, Quick sort)	T2_17 (pg.497)	**CO1-CO6**	
37	Randomized Algorithm: Contention Resolution	T1_13.1 (pg.662-664)	**CO5**	
38	Randomized Algorithm: Contention Resolution	T1_13.1 (pg.665-667)	**CO5**	
39	Randomized Algorithm: Median Finding and Quick Sort	T1_13.5 (pg.681-684)	**CO5**	
40	Randomized Algorithm: Median Finding and Quick Sort	T1_13.5 (pg.684-687)	**CO5**	**Assignment-4**
41	Randomized Algorithm: Hashing	T1_13.6 (pg.691-692)	**CO5**	
42	**Lab#10:** Java implementation of Dynamic Programming based algorithms (Weighted Intrval Scheduling, Longest Common Subsequence, Coin Exchange, Matrix Chain Multiplication)	T2_18 (pg.506)	**CO1-CO6**	
43	Randomized Algorithm: Hashing	T1_13.6 (pg.692-694)	**CO5**	**Quiz-3**

Sl.No.	Lessons/Topics to be covered	Book Reference (sections)	Mapping with COs	Home Work/ Assignments/ Quizzes
44	**Lab#11:** Java implementation of Dynamic Programming based algorithms (Weighted Intrval Scheduling, Longest Common Subsequence, Coin Exchange, Matrix Chain Multiplication)	T2_18 (pg.506)	**CO1-CO6**	
45	How to Design Algorithms; A catalog of Algorithmic Problems	P2_10,11 (pg.363-366)	**CO6**	
46	How to Design Algorithms; A catalog of Algorithmic Problems	P2_10,11 (pg.363-366)	**CO6**	
47	How to Design Algorithms; A catalog of Algorithmic Problems	P2_10,11 (pg.363-366)	**CO6**	
48	**Lab#12:** Java implementation of Dynamic Programming based algorithms (Weighted Intrval Scheduling, Longest Common Subsequence, Coin Exchange, Matrix Chain Multiplication)	T2_18 (pg.506)	**CO1-CO6**	

✡ Few Groups will be assigned to conduct Module based Experiments (Verification / Understanding domain knowledge) and Some groups need to be assigned with term projects (Analysis/ Implementation).