MANIPAL UNIVERSITY JAIPUR

Department of Artificial Intelligence & Machine Learning

Course Hand-out

A. Basic Details:

Programme Name:	B. Tech (AIML)
Course Name:	Design & Analysis of Algorithms
Course Code:	AIM2201
LTPC *(Lecture Tutorial Practical Credits):*	3 1 0 4
Session:	Jan- May 2025
Class:	IV Semester
Course Coordinator:	Mr. Siddharth Kumar
Course Instructor(s):	Dr. Gautam Kumar | Dr. Manish Rai | Dr. Shail Saharan | Dr Kanwal Preet Kour| Dr. Jyoti Rani | Dr. Neha Mahala| Mr. Shivendra Dubey
Additional Practitioner(s) – if any *(Industry Fellow/ Visiting Faculty/ Adjunct Faculty, etc.):*	To be identified and appointed later

B. Introduction:

The main objective of this course is to discuss various techniques for designing efficient algorithms and analyse their complexity and performance. The course is intended to provide the students an experience in algorithm design and to emphasize both the practical as well as the mathematical formulation including the mentioned points.

a. Analyse the asymptotic performance of the designed algorithms.
b. Write correctness proofs for algorithms.
c. Demonstrate familiarity with major algorithms and data structures.
d. Apply important algorithmic design paradigms and methods of analysis.
e. Demonstrate efficient algorithms in common engineering design situations.

C. Course Outcomes: At the end of the course, students will be able to

CO Statement	*CO*	*Level*	*Target Attainment %*	*Target Attainment level*
Find the running times of algorithms using asymptotic analysis.	**AIM2201.1**	**Remembering**	≥ 80%	3
Demonstrate and Design algorithms using the divide-and-conquer paradigm to solve business	**AIM2201.2**	**Understanding**	≥ 80%	3

problems, hence enhancing analytical skills.				
Apply the concept of greedy and dynamic-programming approaches to solve real-life problems to enhance entrepreneurship capabilities.	**AIM2201.3**	**Applying**	≥ 70% < 80%	2
Analyze the concept of backtracking and branch & bound algorithms.	**AIM2201.4**	**Analyzing**	≥ 70% < 80%	2
Determine and **evaluate** various advanced algorithms concepts such as graphs, string matching, approximation algorithms, and complexity classes to enhance employability and entrepreneurship.	**AIM2201.5**	**Evaluating**	≥ 70% < 80%	2

D. PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

[PO.1] **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.

[PO.2] **Problem Analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

[PO.3] **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

[PO.4] **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

[PO.5] **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

[PO.6] **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

[PO.7] **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

[PO.8] **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practices.

[PO.9] **Individual and Teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

[PO.10] **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

[PO.11] **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

[PO.12] **Life-long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES [PSO]

[PSO.1] Graduates will be able to examine the applications of Artificial Intelligence and Machine Learning in real-life problems.

[PSO.2] Graduates will be able to design and implement intelligent systems for multidisciplinary problems.

E. Assessment Plan:

Criteria	Description	Maximum Marks
Internal Assessment (Summative)	Mid-Term Examination (Closed Book)	30
	NPTEL Course Certification (25 Marks) + Attendance (5 Marks) Course Link: https://onlinecourses.nptel.ac.in/noc25_cs23/preview	30
End Term Exam (Summative)	End Term Exam (Closed Book)	40
	Total	100
Attendance (Formative)	A minimum of 75% Attendance is required to be maintained by a student to be qualified for taking up the End Semester examination. The allowance of 25% includes all types of leaves including medical leaves.	
Make up Assignments (Formative)	Students who miss a class will have to report to the teacher about the absence. A makeup assignment on the topic taught on the day of absence will be given which has to be submitted within a week from the date of absence. No extensions will be given on this. The attendance for that day of absence will be marked blank, so that the student is not accounted for absence. These assignments are limited to a maximum of 5 throughout the entire semester.	
Homework/ Home Assignment/ Activity Assignment (Formative)	There are situations where a student may have to work in home, especially before a flipped classroom. Although these works are not graded with marks. However, a student is expected to participate and perform these assignments with full zeal since the activity/ flipped classroom participation by a student will be assessed and marks will be awarded.	

F. Syllabus

Introduction: Algorithm Definition and Criteria of Algorithms, Iterative and Recursive Algorithms. **Performance Analysis:** Priori and Posteriori Analysis, Asymptotic Notations, Space Complexity, Time Complexity, Performance measurement of iterative and recursive algorithms. **Solving Recurrence Relations:** Substitution Method, Iterative Method, Recursive Tree Method, Master Method. **Divide and Conquer:** Introduction, Binary Search, Finding Maximum and Minimum, Merge Sort, Quick Sort, Randomized Quick Sort, Closest Pair of Points, Integer Multiplication. **Fast Fourier Transformers Graph Search Algorithm:** Graph representation, Breadth First Search, and Depth First Search. **Greedy Strategy:** Introduction, Knapsack Problem, Job Sequencing with Deadlines, Huffman Coding, Union and Find Operation (Set and Disjoint Set), Minimum Cost Spanning Tree Algorithms (Prim's and Kruskal's), Optimal Merge Patterns, Single Source Shortest Path (Dijkstra's Algorithm). **Dynamic Programming:** Introduction, Single Source Shortest Path (Bellman and Ford Algorithm), All Pair Shortest Path (Floyd Wrashal's Algorithm), Optimal Binary Search Trees, 0/1 Knapsack Problem, Travelling Salesperson Problem, Longest Common Subsequence, Matrix Chain Multiplication, Edit distance, Viterbi Algorithm. **Backtracking:** Introduction, N-Queens Problem, Graph Colouring, and Hamiltonian Cycles. **Branch and Bound:** Introduction, FIFO and LC Branch and Bound, 0/1 Knapsack Problem, Travelling Salesman Problem. **String Matching:** Naïve String Matching, Rabin Karp Algorithm, Knuth-Morris-Pratt Algorithm, Boyer-Moore Algorithm. **Complexity Classes:** NP, NP-Complete and NP-Hard Problems, Cook's Theorem, Polynomial-time reductions, Satisfiability, Reduction from Satisfiability to Vertex Cover.

Textbooks:

1. Horowitz, S. Sahni, and S. Rajasekaran, *"Computer Algorithms"*, 2nd Edition, University Press, 2008.
2. T. H. Cormen, C. E. Leiserson, R.L. Rivest, and C. Stein, *"Introduction to Algorithms"*, 3rd Edition, MIT Press, 2010.

Reference Books:

1. V. Aho, J. E. Hopcroft and J. D. Ullman, "*The Design and Analysis of Computer Algorithms*", 1st Edition, Pearson Education, 2002.

Web Reference (E-Resources of MUJ):

1) Levitin, A. (2007). *Introduction to the design & analysis of algorithms*. Pearson Education India.

G. Lecture Plan:

Lecture No	Topics	Session Outcome	Corresponding CO	Mode of Delivery	Mode of Assessin CO
1	**Introduction and Course Hand-out Briefing**	Understanding Fundamental Algorithms	2201.1	Lecture Based Teaching Learning	NA
2	Introduction: Algorithm Definition and Criteria of Algorithms, Iterative and Recursive Algorithms.	Understanding Fundamental Algorithms	2201.1	Lecture Based Teaching Learning	MTE, ETE
3	**Performance Analysis:** Priori and Posteriori Analysis	Understanding Fundamental Algorithms	2201.1	Lecture Based Teaching Learning	MTE, ETE
4	Asymptotic Notations	Understanding Fundamental Algorithms	2201.1	Lecture Based Teaching Learning	MTE, ETE
5	Asymptotic Notations	Understanding Fundamental Algorithms	2201.1	Lecture Based Teaching Learning	MTE, ETE
6	Time & Space Complexity	Problem Solving Skills	2201.1	Lecture Based Teaching Learning	MTE, ETE
7	Mathematical Analysis of Recursive Techniques	Problem Solving Skills	2201.1	Lecture Based Teaching Learning	MTE, ETE
8	Mathematical Analysis of non-recursive techniques	Problem Solving Skills	2201.1	Learning through problem solving	MTE, ETE

9	**Solving Recurrence Relations**: Substitution Method, Iterative Method, Recursive Tree Method, Master Method	Algorithm Design Techniques & Algorithm Analysis	2201.2	Learning through problem solving	MTE, ETE
10	Divide and Conquer: Introduction, Binary Search, Finding Maximum and Minimum	Algorithm Design Techniques & Algorithm Analysis	2201.2	Flipped Classroom	MTE, ETE
11	Merge Sort and Analysis	Algorithm Design Techniques & Algorithm Analysis	2201.2	Lecture Based Teaching Learning	MTE, ETE
12	Quick Sort and Analysis	Algorithm Design Techniques & Algorithm Analysis	2201.2	Lecture Based Teaching Learning	MTE, ETE
13	Randomized Quick Sort	Algorithm Design Techniques & Algorithm Analysis	2201.2	Lecture Based Teaching Learning	MTE, ETE
14	Closest Pair of Points	Algorithm Design Techniques & Algorithm Analysis	2201.2	Lecture Based Teaching Learning	MTE, ETE
15	Integer Multiplication	Algorithm Design Techniques & Algorithm Analysis	2201.2 & 2201.5	Lecture Based Teaching Learning	MTE, ETE
16	Graph Search Algorithm: Graph Representation	Graph Algorithms	2201.2	Peer Teaching	MTE, ETE
17	Graph Search Algorithm - BFS/ DFS	Graph Algorithms	2201.3	Flipped Classroom	MTE, ETE
18	Greedy Strategy: Introduction, Knapsack Problem	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
19	Job Sequencing with deadline	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
20	Optimal Merge tape, Huffman Coding	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
21	Spanning Trees - MST (Prim's and Kruskal's Algorithm)	Greedy Algorithms	2201.3	Lecture Based	MTE, ETE

				Teaching Learning	
22	Union and Find Operation (Set and Disjoint Set)	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
23	Optimal Merge Patterns	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
24	Dijkstra's Algorithm-SSSP	Greedy Algorithms	2201.3	Lecture Based Teaching Learning	MTE, ETE
25	Dynamic Programming: Introduction to DP	Dynamic Programming	2201.3	Lecture Based Teaching Learning	MTE, ETE
26	Bellman Ford Algorithm – SSSP	Dynamic Programming	2201.3	Lecture Based Teaching Learning	MTE, ETE
27	Optimal Binary Search Trees	Dynamic Programming	2201.3	Lecture Based Teaching Learning	MTE, ETE
28	0/1 Knapsack Problem	Dynamic Programming	2201.3	Lecture Based Teaching Learning	MTE, ETE
29	Longest Common Subsequence	Dynamic Programming	2201.3	Lecture Based Teaching Learning	MTE, ETE
		Mid Term Examination			
30	Multi-Stage Graph	Dynamic Programming	2201.4	Lecture Based Teaching Learning	MTE, ETE
31	Floyd Warshal Algorithm – All pair of shortest paths	Dynamic Programming	2201.4	Lecture Based Teaching Learning	MTE, ETE
32	Matrix Chain Multiplication	Dynamic Programming	2201.4	Lecture Based Teaching Learning	MTE, ETE
33	TSP-Dynamic Programming	Dynamic Programming	2201.4	Lecture Based	MTE, ETE

				Teaching Learning	
34	Edit distance	Dynamic Programming	2201.4	Lecture Based Teaching Learning	MTE, ETE
35	Viterbi Algorithm	Dynamic Programming	2201.4	Lecture Based Teaching Learning	MTE, ETE
36	Backtracking: Introduction,	Backtracking and Branch Bound	2201.4	Lecture Based Teaching Learning	MTE, ETE
37	N-Queens Problem	Backtracking and Branch Bound	2201.4	Technology based learning	MTE, ETE
38	Graph Colouring	Backtracking and Branch Bound	2201.4	Technology based learning	MTE, ETE
39	Hamiltonian Cycles Problem	Backtracking and Branch Bound	2201.4	Lecture Based Teaching Learning	MTE, ETE
40	Branch and Bound: Introduction	Backtracking and Branch Bound	2201.4	Lecture Based Teaching Learning	MTE, ETE
41	FIFO and LC Branch and Bound	Backtracking and Branch Bound	2201.5	Lecture Based Teaching Learning	MTE, ETE
42	Branch and Bound – 0/1 Knapsack Problem	Backtracking and Branch Bound	2201.5	Lecture Based Teaching Learning	MTE, ETE
43	Branch & Bound – TSP	Backtracking and Branch Bound	2201.5	Lecture Based Teaching Learning	MTE, ETE
44	String Matching: Naïve String Matching, Rabin Karp Algorithm	Problem-Solving Strategies	2201.5	Lecture Based Teaching Learning	MTE, ETE
45	Knuth-Morris-Pratt (KMP) Algorithm	Problem-Solving Strategies	2201.5	Lecture Based Teaching Learning	MTE, ETE
46	Boyer- Moore Algorithm	Problem-Solving Strategies	2201.5	Lecture Based	MTE,

				Teaching Learning	ETE
47	Complexity Classes: NP, NP-Complete and NP-Hard Problems	Problem-Solving Strategies	2201.5	Lecture Based Teaching Learning	MTE, ETE
48	Cook's Theorem	Problem-Solving Strategies	2201.5	Lecture Based Teaching Learning	MTE, ETE
49	Polynomial-time reductions, Satisfiability	Problem-Solving Strategies	2201.5	Lecture Based Teaching Learning	MTE, ETE
End Term Examination					

H. Course Articulation Matrix: (Mapping of COs with POs and PSOs)

CO	STATEMENT	Correlation With Program Outcomes and Program Specific Outcomes													
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
AIM2201.1	Find the running times of algorithms using asymptotic analysis.	1	3		1	1			2					3	
AIM2201.2	Demonstrate and Design algorithms using the divide-and-conquer paradigm to solve business problems hence enhancing skills.	2	2	3				1				2	2		2
AIM2201.3	Apply the concept of greedy and dynamic-programming approaches to solve real-life problems to enhance entrepreneurship capabilities.	2	2	3	2	1							1	2	1
AIM2201.4	Analyse the concept of backtrackingand branch & bound algorithms.				2	1			1					3	2
AIM2201.5	Determine and evaluate various advanced algorithms concepts such as graphs, stringmatching, approximation algorithms, and complexity classes to enhanceemployability.			2										1	

Siddharth Kr

Course Coordinator
Name: Mr. Siddharth Kumar
MUJ ID: MUJ 1693

Head of the Department

Student Representative
Name:
Registration No.