

Semester: 4				
Programme : B.Sc. Computer Science (Hons)				
Course : DISCRETE STRUCTURES				
Paper code: C2CS230411T				Credits: 4
Hours/week : Theory: 4				
Category: Core/MDC/SEC/VAC : Core				
Theory / Practical / Composite : Theory				
No of Modules : 2				
<p>Course Overview: This foundational course bridges abstract mathematics and practical computer science by equipping students with formal tools to model, analyze, and solve computational problems. Students master discrete structures (sets, relations, graphs), logical reasoning (propositional logic), combinatorial techniques (counting, induction), and algorithmic analysis (recurrences, graph algorithms). The curriculum emphasizes mathematical rigor for proving correctness and algorithmic thinking for designing efficient solutions—essential for advanced courses in algorithms, theory of computation, cryptography, and AI.</p>				
Course Outcome:				
1. Apply set theory, relations, functions, and counting principles (Pigeonhole, Inclusion-Exclusion, Induction) to model computational constraints and prove combinatorial properties.				
2. Formulate recurrence relations for recursive algorithms and solve them using substitution, characteristic equations, generating functions, and the Master Theorem to derive asymptotic complexity bounds.				
3. Construct and transform propositional logic expressions into normal forms, and validate logical arguments using equivalences and tautology checking for formal verification tasks.				
4. Analyze graph-theoretic properties (connectivity, planarity, coloring) and implement graph algorithms (BFS/DFS, Dijkstra, MST) to solve optimization and traversal problems in networked systems.				
Prerequisites:				
<ul style="list-style-type: none"> • Basic programming proficiency • High school algebra and logical reasoning • Recommended: Familiarity with basic data structures (arrays, linked lists, stacks, queues) 				
SYLLABUS				
UNIT/Module	CONTENT	HOURS or NUMBER OF CLASSES	CO Mapping	COGNITIVE LEVEL
Group A				
I.	Basic Structures: Sets, Relations, Functions	9	CO1	Apply, Analyse (K3, K4)
II.	Counting Theory: Pigeonhole Principle; Mathematical Induction, Principle of Inclusion and Exclusion	7	CO1	Apply, Analyse (K3, K4)
III.	Recurrence Relations: Substitution Method, Linear Recurrence Relations with constant coefficients and their	10	CO2	Apply, Analyse (K3, K4)

	solution, Generating functions, Recurrence Trees, Master Theorem			
Group B				
IV.	Propositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Normal Forms	6	CO3	Apply, Analyse (K3, K4)
V.	Graph Theory: Definition of Graph, Graph Terminology, Finite and Infinite graphs. Directed and undirected graphs, Degree, Isolated vertex, Pendant vertex. Null graphs. Walks: Paths and circuits. Connected and disconnected graphs, Euler's graphs, Hamiltonian paths and circuits. Planar Graph, Isomorphic graph, Graph Coloring Problem.	10	CO4	Analyse, Create (K4, K6)
VI.	Graph Algorithms: Graph representation, Floyd's shortest Path algorithm, Trees, Minimal Spanning Tree using Kruskal Algorithm and Prim's Algorithm, Dijkstra's Algorithm, Breadth First Search (BFS), Depth First Search (DFS), Connected components.	10	CO4	Analyse, Create (K4, K6)
Text Books				
1. C.L. Liu , D.P. Mahopatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985				
2. Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006				
3. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms , John wiley Publication, 1988				
4. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009				
5. D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers				
Evaluation				
Theory CIA: 25 Attendance: 5 Semester Exam: 70				
Paper Structure for Theory Semester Exam Module: Answer 5 out of 7 of 7 marks each in Group A Answer 5 out of 7 of 7 marks each in Group B				

Course outcomes (COs) and Cognitive Level Mapping

COs	CO Description	Cognitive levels
CO1	Apply set theory, relations, functions, and counting principles (Pigeonhole, Inclusion-Exclusion, Induction) to model computational constraints and prove combinatorial properties.	Apply, Analyse (K3, K4)

CO2	Formulate recurrence relations for recursive algorithms and solve them using substitution, characteristic equations, generating functions, and the Master Theorem to derive asymptotic complexity bounds.	Apply, Analyse (K3, K4)
CO3	Construct and transform propositional logic expressions into normal forms, and validate logical arguments using equivalences and tautology checking for formal verification tasks.	Analyse (K3, K4)
CO4	Analyze graph-theoretic properties (connectivity, planarity, coloring) and implement graph algorithms (BFS/DFS, Dijkstra, MST) to solve optimization and traversal problems in networked systems.	Analyse, Create (K4, K6)