Course Title	**Discrete Structures**		
Course Code	**GE-161**		
Credit Hours	3 (3,0)		
Category	General Education		
Prerequisite	None		
Co-Requisite	None		
Follow Up	None		
Course Introduction	This course provides a foundation in discrete mathematical structures essential to computer science, covering formal logic, set theory, functions, relations, proof techniques, counting, induction, graphs, and trees. It emphasizes mathematical reasoning, abstraction, and problem-solving to model and analyze computational systems, while linking theory to applications in algorithms, data structures, databases, artificial intelligence, and software correctness.		
Course Learning Outcomes (CLOs)	At the end of the course, the students will be able to:	**BT**	**PLO**
	CLO1: Explain fundamental concepts of discrete structures, including logic, sets, functions, relations, graphs, and trees.	C2 (Understand)	1,3
	CLO2: Apply logical reasoning, proof techniques, and discrete structures to model and solve computing problems.	C3 (Apply)	1,2,3,4
	CLO3: Analyze and differentiate discrete structures and evaluate their relevance in computer science applications.	C4 (Analyze)	1,2,3,4
Text Book(s)	Discrete Mathematics and Its Applications 8th Edition by Kenneth Rosen., McGraw Hill Education, 2018. ISBN: 1260091996, 978-1260091991		
Reference Material	Discrete Mathematical Structures 6th Edition by Bernard Kolman, Robert Busby, Sharon C. Ross., Pearson, 2008. ISBN 0132297515, 978-0132297516		

Weekly Lesson Plan

Week No.	Course Content	Assignments/Readings
Week 1	Course introduction and motivation. Importance of Discrete Structures in Computer Science. Applications in algorithms, databases, artificial intelligence, and software engineering. Overview of course contents and learning strategies. Introduction to propositional logic: propositions, truth values, logical connectives, and compound propositions.	Rosen: Chapter 1
Week 2	Construction and evaluation of truth tables. Logical implication, biconditional, and translation of natural language statements into symbolic form. Logical equivalence, tautology, contradiction, and contingency.	Rosen: Chapter 1
Week 3	Logical equivalence using truth tables and standard laws of logic. Simplification of logical expressions. Introduction to predicate logic: predicates, propositional functions, and domain of discourse.	Rosen: Chapter 1–2 Quiz1
Week 4	Universal and existential quantifiers. Negation of quantified statements. Nested quantifiers and interpretation in natural language.	Rosen: Chapter 2

Week 5	Set theory fundamentals: sets, universal set, subsets, cardinality. Power sets and properties. Set operations: union, intersection, difference, and complement. Algebra of sets and identities.	Rosen: Chapter 2 Quiz2
Week 6	Set equality and verification of set identities using Venn diagrams and membership tables. Characteristic vectors and representation of sets as bit-vectors. Applications of sets in computing.	Rosen: Chapter 2 Quiz3
Week 7	Functions: ordered pairs, tuples, and Cartesian products. Definition and representation of functions. Types of functions (injective, surjective, bijective). Inverse and numeric functions with applications.	Rosen: Chapter 2–3 Quiz4
Week 8	Relations: definition and notation. Representation using matrices and directed graphs. Properties of relations: reflexive, symmetric, antisymmetric, and transitive.	Rosen: Chapter 9
Week 9	Composition of relations and closures. Equivalence relations and equivalence classes. Partitioning of sets. Partial order relations and posets.	Rosen: Chapter 9
Week 10	Sequences and summations. Arithmetic and geometric progressions. Recursive definitions. Summation notation, linearity, and evaluation of common and geometric sums.	Rosen: Chapter 3 Quiz5
Week 11	Counting principles: introduction and applications. Sum rule, product rule, complement rule, and inclusion–exclusion principle.	Rosen: Chapter 6 Quiz6
Week 12	Permutations and combinations. Binomial coefficients and applications in counting problems.	Rosen: Chapter 6
Week 13	Mathematical proofs: rules of inference. Direct proof, proof by contrapositive, and proof by contradiction.	Rosen: Chapter 1–3 Quiz7
Week 14	Mathematical induction: principle of induction and applications. Strong induction. Introduction to graph theory: terminology, types, Handshaking Lemma, and graph representations.	Rosen: Chapter 5
Week 15	Walks, paths, and cycles. Connected and strongly connected graphs. k-connected graphs and degree concepts. Havel–Hakimi theorem.	Rosen: Chapter 5 Quiz8
Week 16	Special classes of graphs: complete graphs, paths, cycles, stars, wheels, n-cubes, and bipartite graphs. Trees and their properties. Applications of graphs in computer science.	Rosen: Chapter 5