<table>
<tr><td colspan="5">Devi Ahilya Vishwavidhyalaya, Indore, India
Institute of Engineering & Technology</td><td>II Year B.Tech.
(Computer Science and Engineering)</td></tr>
<tr><td>Course Code & Name</td><td colspan="5">Instructions Hours per Semester and Credits</td></tr>
<tr><td rowspan="2">3RCBS1
DISCRETE MATHEMATICS</td><td colspan="2">Classroom Instruction (CI)</td><td>Lab Instruction (LI)</td><td>Term Work (TW) and Self Learning (SL)</td><td>Total no. of Hours Per semester</td><td>Total Credits (Total Hours/30)</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>TW+SL</td><td rowspan="2">90</td><td rowspan="2">3</td></tr>
<tr><td>30</td><td>10</td><td>0</td><td>50</td></tr>
</table>

Learning Objectives:

1. To introduce students to ideas and techniques from discrete mathematics that are widely used in Computer Science.
2. Provide the fundamentals of formal techniques for solving the problems in mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modelling.

Prerequisites: Nil

COURSE CONTENTS

Unit-I

Sets: Computer representation, principle of inclusion- exclusion.
Functions: floor and ceiling functions**,** mod function, hashing function.
Relations: Binary Relation, equivalence and partial order relations, Hasse Diagram, totally ordered set, lattices, properties of lattices, topological sorting.

Unit-II

Propositional Logic: Proposition, logical equivalence, tautology, contradiction and contingency, normal forms, logic in proof, predicates and quantifiers, rules of inferences, Applications to logical puzzles, Boolean searches, logic circuits and system specifications.

Unit-III

Mathematical Induction: Strong induction, well-ordering, Recursive definitions.
Advanced Counting techniques: Pigeon-hole principle, generating functions.
Recurrence relations: -Formation, methods of solution and application.

Unit-IV

Graph Theory: Graph representation-incidence and adjacency matrices, connectedness, graph isomorphism, Euler & Hamiltonian graphs, planar graph, graph coloring, matching, shortest paths algorithms.
Trees: Spanning trees, tree traversals; prefix codes.

Unit-V

Groups-Monoids, Groups, cyclic group, definition and examples of Rings, Integral domain and Fields; Application to coding theory.
Boolean Algebra-Laws of Boolean algebra, Boolean functions, normal forms, simplification of Boolean function by Karnaugh Map Method, Boolean expression for logic and switching network.

Course Outcome (CO):

CO. No.	CO
CO1	Inclusion-exclusion principle helps to compute the number of elements that satisfy at least one of several properties when elements satisfying more than one property are not counted twice. Functions provides a relation between a set of inputs and a set of permissible outputs. Relations will help in establishing a connection between any two objects or things and in creating relational database management systems.
CO2	Propositional Logic can be used in validating arguments and decision making.
CO3	Mathematical induction is a technique to prove algorithm correctness, pigeon hole principle in counting, generating function to manipulate the sequence as a single entity, recurrence relation in analysing and solving problems related to algorithms, signal processing, control systems, and computational complexity.
CO4	Graphs helps to model real problems, study and analysis. Application of trees is in data retrieval, data compression, representing hierarchical relationships, design of networks, etc.
CO5	Algebraic structures provide a framework for constructing systems and analysing their properties. They provide rules, how elements interacts and create new elements. Boolean algebra has its applications in development of digital electronics, digital logic, computer programming, and mathematical logic.

Books Recommended:

[1]. Kenneth H. Rosen, Discrete Mathematics and its Applications, 7th ed., Tata McGraw-Hill Edition 2007.
[2]. Kolman, Busby & Ross, Discrete Mathematical Structures, 6th edition, Pearson Education, 2008.
[3]. C.L. Liu, Introduction to Discrete Mathematics, McGraw Hill, 1986.
[4]. Trembley and Manohar, Discrete Mathematical structures for Computer Science, McGraw Hill, 1986.
[5]. Edgar G. Goodaire, Michael M. Parmenter, Discrete Mathematics with Graph Theory, 3rd edition Prentice Hall, 2005.

CO-PO-PSO Relationship

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
3RCBS1.CO1	3	3	2	2	3							2	3	1
3RCBS1.CO2	3	3	2	2	3							2	3	1
3RCBS1.CO3	3	3	2	2	3							1	3	2
3RCBS1.CO4	3	3	2	2	3							2	3	2
3RCBS1.CO5	3	3	2	2	3							2	2	3