GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Master of Computer Applications

Level: Post Graduate

Course / Subject Code:

Course / Subject Name: Basic Mathematics

w. e. f. Academic Year:	2024-25
Semester:	1
Category of the Course:	Non-Credit Compulsory Subject for the students who did not have mathematics at undergraduate level.

Prerequisite:	High school mathematics, Basic understanding of algebra and proficiency in problem-solving skills.
Rationale:	This course is designed to equip students with foundational knowledge across various disciplines. Starting with Set Theory & Fundamentals, students learn to manipulate sets and matrices, crucial for applications in database management and algorithms. Mathematical Logic builds on this by teaching rigorous reasoning and logical argument construction, skills vital for software development and formal verification. Integers and Counting deepen students' understanding of concepts like prime numbers and counting techniques, which are essential for cryptography and data science. Relations and Functions introduce students to modeling relationships and mappings applicable to network analysis and machine learning. Lastly, Graphs and Trees provide tools to analyze complex systems, preparing students to solve social network and data structure problems. This structured approach ensures students develop theoretical understanding and practical skills necessary for future academic and professional success in technology-driven fields.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Apply set theory concepts to solve problems involving set operations, describe sets using roster and set-builder notation, and manipulate matrices through various operations.	AP
02	Construct truth tables for logical expressions, apply DeMorgan’s laws to simplify logical statements, and use mathematical induction to prove statements about integers.	AP
03	Calculate greatest common divisors (GCD) and least common multiples (LCM), convert between different numeral systems, and solve counting problems using permutations, combinations, and recurrence relations.	AP
04	Define and classify relations based on their properties, analyze functions for injectivity, surjectivity, and bijectivity, and apply composition and inverse operations to functions.	AP

05	Interpret and construct different types of graphs, analyze paths and cycles within graphs, identify and apply properties of trees, and perform tree traversals using various methods.	AP

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	P	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
0	2	2	0	0	50	50	0	100

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Unit 1: Set Theory & Fundamentals Set Theory** Basic Concepts of Set Theory: Definition, Two Methods to Describe (Represent) Sets; Sets and Subsets, Operations on Sets, Algebraic Properties of Set Operations, The Addition Principle, Sequences: Introduction, Characteristic Functions, Strings and Regular Expressions Matrices: Representation of a Matrix; Equality of Matrices; Special Matrices: Rectangular / Square Matrices, Null (Zero) Matrix, Unit Matrix, Diagonal Matrices, Sum and Difference of 2 Matrices; Multiplication of 2 matrices; Transpose of a Matrix, Symmetric Matrices, Boolean Matrix Operations	5	15
2.	**Unit 2: Mathematical Logic** Introduction, Propositions and Logical Operations, Truth Tables; DeMorgans' Laws; Conditional Statements, Methods of Proof, Mathematical Induction, Mathematical Statements	7	25
3.	**Unit 3: Integers and Counting** Integers: Properties of Integers; Prime Number; Greatest Common Divisor (GCD); Relative Prime; Least Common Multiple (LCM); Representation of Integers in Computer; Decimal, Binary, Octal, and Hexadecimal Representation Counting: Permutations, Combinations, Pigeonhole Principle, Elements of Probability, Recurrence Relations	6	20
4.	**Unit 4: Relations and Functions** Relations: Definition, Binary Relation, Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection, and	8	25

	Complement Operations on Relations, Properties of Binary Relations in a Set: Reflexive, Symmetric, Transitive, Antisymmetric Relations, Relation Matrix and Graph of a Relation; Partition and Covering of a Set, Equivalence Relation, Equivalence Classes, Compatibility Relation, Maximum Compatibility Block, Composite Relation, Converse of a Relation, Transitive Closure of a Relation R in Set X Functions: Introduction & Definition, Argument. Co-domain, Range, Image, Value of a Function; Examples, Peano's Successor Function; Onto (surjective), Into, One-to-one (injective), Many- to-one, Bijective (one-to-one and onto); examples; Composition of Functions, examples; Inverse Function, Identity Map, Condition of a Function to be Invertible, examples; Inverse of Composite Functions, Properties of Composition of Functions; Binary and n-ary Operations as Mappings (functions), Properties of Binary Operations; Characteristic Function of a Set; Properties, examples; Hashing Functions: Division Method, and Mid-square Method, examples; *Note: Proofs of Theorems are not required*		
5.	**Unit 5: Graphs and Trees** Graphs: Introduction, Definition; Initial & Terminal Nodes, Adjacent Nodes; Directed Edge, Undirected Edge, Directed Graph (Digraph), Undirected Graph, Mixed Graph; Loop (Sling); Distinct Edges, Parallel Edges; Multi-graph, Simple Graph; Weighted Graph; Isolated Nodes, Null Graph; Isomorphic Graphs; In-degree, Out-degree, Total-degree; Subgraphs; Reflexive, Symmetric, Transitive Digraphs; Paths, Length of Path of a Graph; Simple Path (Edge Simple), Elementary Path (Node Simple), Cycle (Circuit), Simple Cycle, Elementary Cycle; Path of Minimum Length (Geodesic), Distance between Two Nodes, Triangle Inequality; Reachability, Reachable Set of a Node, Connected Graphs: Strongly, Unilaterally, Weakly Connected Graphs & Components; Trees: Introduction, Definition, Root, Branch Nodes, Leaf (Terminal Node); Different Representations of Trees; Forests, Subtrees; M-ary Tree, Full or Complete M-ary Tree; Binary Tree, Full (Complete) Binary Tree; Conversion of M-ary Tree to Binary Tree; Traversal of Binary Tree: Pre-order, In-order, and Post-order Traversal *Note: Proofs of Theorems are not required*	4	15
	Total	**30**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
5	**10**	**85**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. J. P. Tremblay and R. Manohar, "Discrete Mathematical Structures with Applications to Computer Science", Tata McGraw-Hill (2010) – only for Unit-5 (Graphs & Trees).
2. Bernard Kolmann & others, "Discrete Mathematical Structure", Pearson Education, Sixth Edition
3. K. H. Rosen, "Discrete Mathematics and its applications", Tata McGraw-Hill, 6th edition
4. D. S. Malik & M. K. Sen, "Discrete Mathematics", Cengage Learning (2004)
5. Edgar G. Goodaire, Michael M. Parmenter. "Discrete Mathematics with Graph Theory", PHI
6. Ralph P Grimaldi & B V Ramana, "Discreet and Combinatorial mathematics: An Applied Introduction", Pearson Education, 5th Edition (2018)
7. J. P. Tremblay and W. K. Grassman. "Logic and Discrete Mathematics", Pearson Education

(b) Open-source software and website:

1. https://www.scilab.org/ (This is an opensource numerical and mathematical simulation tool, students may be given some exercise on using this tool to solve the problems)

Suggested Course Tutorial List:

Following tutorial assignments in each unit should be planned

Unit 1: Set Theory & Fundamentals

1.1 Basic Concepts of Set Theory
- Tutorial Assignments: Solve problems on set operations and properties

1.2 Sequences
- Tutorial Assignments: Analyze given sequences and regular expressions

1.3 Matrices
- Tutorial Assignments: Homework: Solve problems involving matrix operations

Unit 2: Mathematical Logic

2.1 Introduction to Mathematical Logic
- Tutorial Assignments: Solve logic problems using truth tables and proofs

Unit 3: Integers and Counting

3.1 Integers
- Tutorial Assignments: Solve problems on prime numbers and integer representations

3.2 Counting
- Tutorial Assignments: Solve problems involving permutations, combinations, and probability

Unit 4: Relations and Functions

4.1 Relations
- Tutorial Assignments: Solve problems involving binary relations and their properties

4.2 Functions
- Tutorial Assignments: Solve problems on function composition and inversion

Unit 5: Graphs and Trees

5.1 Graphs
- Tutorial Assignments: Solve problems involving graph properties and paths

5.2 Trees

Tutorial Assignments: Solve problems involving tree traversal and representation

Tips for Effective Tutorials

- **Use multimedia:** Incorporate videos, animations, and interactive tools to explain complex concepts.
- **Encourage participation:** Use discussion forums, peer reviews, and group projects to engage students.
- **Provide feedback:** Offer constructive feedback on assignments and activities to help students improve.

Suggested Project List:

Form groups of students. Each group consists of two students. Ask them to write C/C++ program on any two projects from the list given below –

Unit 1: Set Theory & Fundamentals

Project 1: Set Operations
- **Description**: Create a program to perform various set operations (union, intersection, difference, complement) on sets represented as arrays.
- **Key Concepts**: Arrays, functions, set operations.

Project 2: Matrix Operations
- **Description**: Develop a matrix calculator that can perform operations such as addition, subtraction, multiplication, and transpose on matrices.
- **Key Concepts**: 2D arrays, nested loops, functions for matrix operations.

Unit 2: Mathematical Logic

Project 3: Truth Table Generator

- **Description**: Write a program that generates the truth table for given logical expressions.
- **Key Concepts**: Logical operators, loops, conditionals.

Project 4: Proof Verification using Logical Operations

- **Description**: Implement a program to verify simple logical proofs and conditional statements.
- **Key Concepts**: Logical operators, functions, conditionals.

Unit 3: Integers and Counting

Project 5: Number Conversion Tool

- **Description**: Create a tool that converts numbers between different bases (decimal, binary, octal, hexadecimal).
- **Key Concepts**: Loops, conditionals, arrays, string manipulation.

Project 6: Prime Number Finder

- **Description**: Write a program to find all prime numbers up to a given number using the Sieve of Eratosthenes algorithm.
- **Key Concepts**: Arrays, loops, functions.

Project 7: Permutations and Combinations Calculator

- **Description**: Develop a program to calculate permutations and combinations for given values of n and r.
- **Key Concepts**: Recursion, loops, functions, factorial calculation.

Unit 4: Relations and Functions

Project 8: Binary Relation Representation

- **Description**: Implement a program to represent and manipulate binary relations using matrices and graphs.
- **Key Concepts**: 2D arrays, graph representation, matrix operations.

Project 9: Function Composition and Inversion

- **Description**: Create a program to perform composition and inversion of mathematical functions represented as arrays or mappings.
- **Key Concepts**: Arrays, functions, recursion.

Unit 5: Graphs and Trees

Project 10: Graph Traversal Algorithms

- **Description**: Write a program to perform depth-first search (DFS) and breadth-first search (BFS) on a graph.

- **Key Concepts**: Graph representation using adjacency matrix/list, recursion, queues, and stacks.

Project 11: Binary Tree Operations

- **Description**: Implement a binary tree with functionalities such as insertion, deletion, and traversal (in-order, pre-order, post-order).
- **Key Concepts**: Structures, pointers, recursion.

Project 12: Shortest Path Finder

- **Description**: Create a program to find the shortest path in a weighted graph using Dijkstra's algorithm.
- **Key Concepts**: Graph representation, priority queues, algorithms.

CO- PO Mapping:

Semester ___	**Course Name : Basic Mathematics**							
	POs							
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	-	-	1	-	-	-	-
CO2	2	-	-	1	-	-	-	-
CO3	2	-	-	1	-	-	-	-
CO4	3	-	-	1	-	-	-	-
CO5	1	-	-	1	-	-	-	-
	1.8	**-**	**-**	**1**	**-**	**-**	**-**	**-**

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

* * * * * * *