GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Basic Mathematics (BM)
Subject Code: 619403

With effective from academic year 2020-21

1. Objective

- The objective of this course is to present the foundations of many basic mathematical topics used in Computer Science including RDBMS, Data Structures, Analysis of Algorithms, Theory of Computation, Cryptography, Artificial Intelligence, Statistics and others.
- This course will enhance the student's ability to think logically and mathematically.

2. **Prerequisites**: School level Mathematics; Binary number system

3. **Contents**:

Sr. No.	Topics *(Proof of Theorems not required)*	Weightage Percentage
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	20%
2	**Mathematical Logic** Introduction, Propositions and Logical Operations, Truth Tables; DeMorgans' Laws; Conditional Statements, Methods of Proof, Mathematical Induction, Mathematical Statements	20%
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	15%
4	**Relations and Functions** Relations: Definition, Binary Relation, Representation, Domain, Range, Universal Relation, Void Relation, Union, Intersection, and 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,	25%

	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; *Proofs of Theorems not required*	
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;	20%

Note : *Proofs of Theorems not required*

4. Text Book:

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

5. Reference Books:

1. K. H. Rosen, “Discrete Mathematics and its applications”, Tata McGraw-Hill, 6th edition,
2. D. S. Malik & M. K. Sen, “Discrete Mathematics”, Cengage Learning (2004)
3. Edgar G. Goodaire, Michael M. Parmenter. “Discrete Mathematics with Graph Theory”, PHI
4. Ralph P Grimaldi & B V Ramana, “Discreet and Combinatorial mathematics: An Applied Introduction”, Pearson Education, 5th Edition (2018)
5. J. P. Tremblay and W. K. Grassman. “Logic and Discrete Mathematics”, Pearson Education

6. Chapter-wise Coverage from the Text Book:

Unit No.	Text Books	Topics/Subtopics
I	Book-2	Chapter 1- 1.1 to 1.3 , 1.5
2	Book-2	Chapter 2 - 2.1 to 2.5
3	Book-2	Chapter 1 - 1.4, Chapter 3 - 3.1 to 3.5
4	Book-1	Chapter 2 - 2-3 (2-3.1 to2-3.7), 2-4 (2-4.1 to 2-4.6)
5	Book-1	Chapter 5 - 5-1 (5-1.1 to 5-1.3), 5-1.4, 5-2

7. Accomplishment of Students after Completing the Course:

Students will be able to understand and apply the concepts of sets, logic, cross product of sets and relation, functions, matrices, and basic algorithms related with binary tree and graphs.

8. Active Learning Assignment

Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work for matrix manipulation problem, which includes

- Introduction
- Flow chart
- Pseudo code
- Explanation

9. Laboratory Exercises

Write programs in C language for the following tasks.

Set Theory:

- Set Theory
  - (a) Start with a NULL set and add elements one-by-one: Use different ways of implementing sets and understand the pros and cons of each of these methods
  - (b) Given an element value, check whether it is a member of the set or not
  - (c) Find out the number of elements of a given set.
  - (d) Complement of a set; Union, Intersection
  - (e) Test whether a given set X is a subset of the set A or not.
  - (f) Test whether two given sets are equal or not
  - (g) Difference and Symmetric Difference of two sets
- Create a set. Make sure that addition of elements does not accept any duplicate element. Assume that all elements of the set will be a non-negative integer < 64[0, 63].

  Hint: Create an array of size 64. Store the element x in index x.
  - Create 2 set A and B of size n_1 and n_2 . Print sets A and B.
  - Find ~A, ~B, $A \cup B$, $A \cap B$, $A - B$, $B - A$, and print the size of each derived set.
  - Find whether an element $x \varepsilon A$; $x \varepsilon B$; Find whether $A \subseteq B$; $B \subseteq A$; etc.

Desirable

- Assume that all elements of the set will be an uppercase letter ['A', 'Z'].
  Hint: Create an array of size 26. Store the element x in index x – 'A'. (Note: Try also with lowercase letters as elements of the set.)
  i. Create 2 set A and B of size n_1 and n_2 . Print sets A and B.
  ii. Find ~A, ~B, A ∪ B, A ∩ B, A – B, B – A, and print the size of each derived set.
  iii. Find whether an element x ε A; x ε B; Find whether A ⊆ B; B ⊆ A; etc.

Propositional Logic:

1. Consider 2 symbols: a and b. Represent various propositions using symbols a and b, such as ~a; ~b; (a ^ b); (a v b); (a => b); (b => a); (a <=> b). Determine the Truth value for each of these propositions for the following options:
   i. a is True and b is True
   ii. a is True and b is False
   iii. a is False and b is True
   iv. a is False and b is False
2. Verify DeMorgan's Laws: Hint: Find the Truth value of the LHS and RHS for 4 cases as mentioned above and compare.
   i. ~(a ^ b) = ~a v ~b
   ii. ~(a v b) = ~a ^ ~b

Matrices:

1. Write a function to Create a Matrix of size m x n, and another function to Print a Matrix of size m x n.
2. Create 2 matrices A and B of size (m x n). Find (A + B) and (A – B).
3. Find Transpose of matrix A and of matrix B. Find ($A^T + B^T$) and ($A^T – B^T$). Check whether ($A^T + B^T$) = $(A + B)^T$ and ($A^T – B^T$) = $(A – B)^T$.
4. Add matrix A and null matrix. Subtract null matrix from matrix A.
5. Create a unit matrix of size (n x n) and a unit matrix of size (n x n). Multiply matrix A with a unit matrix.
6. Create matrix A of size (m x n) and matrix B of size (n x p). Multiply matrix A and matrix B to get matrix C of size (m x p).
7. Create a symmetric matrix A. Find matrix A^T . Check whether A = A^T ?
8. Evaluate Scalar Product of a Matrix A: For example, k A, where k is a constant (number)
9. Take as input two matrices, A & B and print (A * B) and (B * A). First check which ones out of (A * B) and (A * B) are possible to compute.

Integers:

- Given a positive integer, find its divisors. Example: Divisors of 36 are 2, 3, 4, 6, 9, 12, 18.
- Given a positive integer n, represent n as product of its divisors. Example: 36 = 2 x 2 x 3 x 3.
- Given a positive integer, find whether it is a prime number or a composite number. Write an efficient algorithm.
- Given two positive integers, m and n, find whether they are relative prime numbers or not.
- Given two positive integers, a and b, find Least Common Multiples (LCM) of a and b.
- Given two positive integers a and b, find GCD (Greatest Common Divisor) of a and b.

- Given a positive integer n, convert it into a corresponding binary number, octal number and hexadecimal number.
- Addition of two binary numbers.
- Subtraction of a smaller binary number from a larger binary number.

Desirable

- Multiplication of a binary number by (a) 2, (b) 4, (c) 8, (d) 16
- Integer division (only quotient will be the answer) of a binary number by (a) 2, (b) 4, (c) 8, (d) 16
- Convert a negative integer into a corresponding binary number.
- Convert a decimal number into a corresponding binary number.
- Subtraction of two binary numbers.
- Multiplication of a binary number by another binary number.

Relations:

1. Input: Two sets A and B and a set of ordered pairs (a, b) such that a ε A, and b ε B. Output: Matrix representation of the Relation.
2. Input: Matrix representation of a relation.
   Output: Whether the relation is (a) Reflexive, (b) Symmetric, (c) Transitive, (d) Anti-symmetric.
3. Input: An Equivalence Relation.
   Output: Equivalence Classes

Functions:

1. Examples of Inverse Function:
   a. Given a positive integer, determine whether it is factorial of some integer or not?
   b. Given a positive integer, find out whether it belongs to Fibonacci sequence or not?
2. Input: Two matrices: Matrix 'a' of size (m x n) and Matrix 'b' of size (n x m).
   1. Output: Find out whether Matrix 'a' is inverse of Matrix 'b' or not?
3. Write functions floor1(), ceiling1() and round1() and test them.

Graphs and Trees :

- Input: Adjacency Matrix of a graph.
  - Output: Print In-degrees and Out-degrees of all nodes
  - Output: Print a sub-graph by removing the 1st node
  - Output: For the 1st node, Print all the paths. Also determine whether there are cycles in the graph? If yes, print the cycles.
- Input: Adjacency Matrix of a binary tree
  - Output: Print whether it is a complete binary tree? Print nodes which do not have 2 children.
  - Output: Print the length of left sub-tree and the length of right sub-tree.
  - Output: Print the sequence of nodes to be traversed during
    (a) Pre-order traversal, (b) in-order traversal, and (c) Post-order traversal