Anekant Education Society's
Tuljaram Chaturchand College of Arts, Science and Commerce, Baramati

Autonomous

Course Structure for B.Sc.(Computer Science) Mathematics

F. Y. B. Sc.(Computer Science) Mathematics

Semester	Paper Code	Title of Paper	No. of Credits
I	CSMT1101	Graph Theory	2
	CSMT1102	Algebra	2
	CSMT1103	Mathematics Practical based on CSMT1101 & CSMT1102	2
II	CSMT1201	Discrete Mathematics	2
	CSMT1202	Calculus	2
	CSMT1203	Mathematics Practical based on CSMT1201 & CSMT1202	2

S. Y. B. Sc.(Computer Science) Mathematics

Semester	Paper Code	Title of Paper	No. of Credits
III	CSMT2301	Linear Algebra	3
	CSMT2302	Numerical Analysis	3
	CSMT2303	Mathematics Practical I	2
IV	CSMT2401	Computational Geometry	3
	CSMT2402	Operations Research	3
	CSMT2403	Mathematics Practical II	2

SYLLABUS (CBCS) FOR F. Y. B. Sc.(COMPUTER SCIENCE)
MATHEMATICS
(w.e.f. June, 2019)

Academic Year 2019-2020

Class **:** F.Y. B. Sc.(Computer Science) (Semester- II)
Paper Code: CSMT1201

Paper **:** I **Title of Paper** **:**Discrete Mathematics
Credit **:** 2 **No. of lectures:** 36

Topics/Contents

Unit 1: Relation and Digraph [9 Lectures]

1.1 Ordered pairs, Cartesian Product of sets.
1.2 Relation, types of relation, equivalence relation, Partial Ordering.
1.3 Equivalence Class, properties and Partition of a set.
1.4 Transitive Closure and Warshall's Algorithm.
1.5 Digraphs of relations ,matrix representation and composition of relations.
1.6 Applications to Computer Science.

Unit 2:Logic [5 Lectures]

2.1 Revision: Propositional Logic, Propositional Equivalences.
2.2 Predicates and Quantifiers: predicate, n-place predicate or n-ary predicate, Quantifications
and Quantifiers, Universal Quantifier, Existential Quantifier, Quantifiers withrestricted domain, Logical Equivalences involving Quantifiers.
2.3 Rules of Inference : Argument in propositional Logic, Validity Argument (Direct and Indirect Methods) Rules of Inference for propositional Logic ,Building Arguments.
2.4 Applications to Computer Science.

Unit 3:Lattices and Boolean Algebra [5 Lectures]

3.1 Poset, Hasse diagram.
3.2 Lattices, Complemented Lattice, Bounded Lattice and Distributive Lattice.
3.3Boolean functions: Introductions, Boolean variable, Boolean function of degree n, Boolean
identities, Definitions of Boolean algebra.
3.4Representations of Boolean functions: Minterm, Maxterm, Disjunctive normal form, Conjunctive normal form.
3.5 Applications to Computer Science.

Unit 4:Recurrence Relations [9 Lectures]

4.1 Recurrence Relations: Introduction, Formation
4.2 Linear Recurrence Relations with constant coefficients.

4.3 Homogeneous solutions
4.4 Particular solutions.
4.5 Total solutions.
4.6 Applications to Computer Science.

Unit 5: Counting Principles [8 Lectures]
5.1 Cardinality of Sets: Cardinality of finite Sets.
5.2 Basics of Counting:The Product Rule, The Sum rule, The Inclusion-Exclusion Principle.
5.3 Generalized Permutations and Combinations.
5.4 Applications to Computer Science.

<u>Text Book</u>:

Kenneth Rosen, Discrete Mathematics and its applications, McGraw Hill Education Pvt. Ltd. (7^{th} Edition)
Chapter – 1, 5, 6, 7, 10.

<u>Reference Books</u>:

1. Discrete Mathematics Structure- Bernard Kolman, Robert Busby, Sharon Culter Ross, Nadeem-ur-Rehman , Pearson Education, 5^{th} Edition.
2. Elements of Discrete Mathematics – C. L. Liu, Tata McGraw Hill.

Class **:** F.Y. B. Sc.(Computer Science) (Semester- II)
Paper Code: CSMT1202

Paper **:** II **Title of Paper** **:**Calculus
Credit **:** 2 **No. of lectures:** 36

Topics/Contents

Unit 1:Continuity and Differentiability [12 Lectures]

1.1 Continuity and Properties of Continuous functions defined on [a, b] (Without Proof) and Examples.
1.2 Differentiability
1.3 Theorem- Differentiability implies continuity but not conversely. Left hand derivative and
Right hand derivative.
1.4 Intermediate value theorem (Without Proof).
1.5 Rolle's theorem (With Proof and Geometric interpretation)
1.6 Lagrange's Mean Value Theorem (With Proof and Geometric interpretation)
1.7 Cauchy's Mean Value Theorem (With Proof) Verification and Application.
1.8 L' Hospital Rule (Without Proof)

Unit 2: Successive Differentiation [6 Lectures]
2.1 The n^{th} derivative of standard functions
2.2 Leibnitz's Theorem (Without Proof)
2.3 Applications to Computer Science.

Unit 3:Taylor's and Maclaurin's Theorems [8 Lectures]
3.1 Taylor's and Maclaurin's Theorems with Lagrange's and Cauchy's form of remainders (Without Proof)
3.2 Taylor's and Maclaurin's Series.
3.3 Applications to Computer Science.

Unit 4:Differential Equation. [10 Lectures]
4.1 Basic Concepts : Introduction, Definition, Directions fields.
4.2 First Order Differential Equation: Linear Differential Equation, separable Differential Equation, Exact Differential Equation, BernoulliDifferential Equation.
4.3 Applications to Computer Science.

<u>Text Books</u>:

1. Shanti Narayan, Mittal, Differential Calculus, S. Chand and Company Ltd, 1998. Chapters – 3, 4, 5.

2. G. F. Simmons, Differential Equations with applications and Historical Notes, Tata Mc-Graw Hill (Second Edition)
   Chapter – 2.

<u>Reference Books</u>:

1. Calculus and Analytical Geometry, Thomas Finny.
2. Differential Equations with Applications and Historical notes, George Simmons.

Class :F.Y. B. Sc.(Computer Science) (Semester- II)
Paper Code: CSMT1203
Paper :III **Title of Paper :** Practical based on CSMT1201 & CSMT1202
Credit :2 **No. of lectures:** 48

Title of Experiments:

Discrete Mathematics:

1 Relations and digraphs
2 Logic
3 Lattices and Boolean Algebra
4 Recurrence Relation
5 Counting Principles
6 Discrete Mathematics using Maxima software
7 Miscellaneous

Calculus:

1 Continuity
2 Differentiability
3 Successive Differentiation
4 Taylors Series
5 Differential Equations
6 Calculus using Maxima Software
7 Miscellaneous

Choice Based Credit System Syllabus (2019 Pattern)

Mapping of Program Outcomes with Course Outcomes

Class: F.Y.B.Sc.(Computer Science). (Sem II)
Subject: Mathematics
Course: Discrete Mathematics **Course Code:** CSMT1201

A) Course Objectives:

1. To understand concepts of Lattices, Boolean algebra, Recurrence relation
2. Use of discrete mathematics in computer science
3. Analyze properties of relations in terms of reflexivity, symmetry, and transitivity.
4. Define equivalence classes and analyze their properties.
5. Apply Warshall's Algorithm to find transitive closures.
6. Recognize and apply concepts of relations and digraphs in computer science scenarios.
7. Apply counting principles to solve problems in computer science.

B) Course Outcome:

1. Express logic sentences in terms of predicates, quantifiers and also evaluate Boolean functions and simplify expression using Boolean algebra.
2. Define and calculate transitive closure in relation to graphs
3. Apply quantifiers to create logical expressions.
4. Apply rules of inference, including direct and indirect methods.
5. Build and evaluate logical arguments using propositional logic.
6. Apply logic concepts to solve problems in computer science.
7. Analyze minterms, maxterms, disjunctive normal form, and conjunctive normal form.

Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	3					2	1
CO 2		3	3				3
CO 3			3				
CO 4		2	1				3
CO 5	1					1	
CO 6		3				1	2
CO 7	3		3			2	1

Justification for the mapping

PO1: Computer Knowledge

CO1: Expressing logic sentences in terms of predicates, quantifiers, and evaluating Boolean functions allows for precise representation and manipulation of logical statements, facilitating effective problem-solving and algorithmic design in computer science through the application of formal logic and Boolean algebra.
CO5: Enhances computational problem-solving skills by enabling systematic analysis and validation of logical structures in software development through the application of propositional logic.
CO7: Analyzing minterms, maxterms, disjunctive normal form (DNF), and conjunctive normal form (CNF) is essential in digital logic design for Boolean algebra manipulation, enabling efficient representation and simplification of logical expressions used in designing digital circuits.

PO2: Design / Development of solution

CO2: Transitive closure in graph theory is essential for determining the reachability between all pairs of vertices, aiding in the comprehensive analysis of connectivity within a graph during the design and development of a solution.
CO4: Applying rules of inference facilitates sound logical reasoning, enhancing the design and development of solutions by ensuring coherent and valid progression from premises to conclusions through direct and indirect methods.
CO6: Applying logic concepts in computer science facilitates the design and development of efficient solutions by ensuring systematic problem-solving, effective decision-making, and optimized algorithmic approaches.

PO3: Modern tool usage

CO2: Transitive closure in graph theory facilitates the determination of all possible paths between vertices, essential for efficient graph analysis and optimization in modern tools.
CO3: Applying quantifiers in modern tools enhances precision and efficiency by expressing logical conditions, enabling more robust and expressive representation of relationships within data and programming environments.
CO4: Applying rules of inference, including direct and indirect methods, enhances logical reasoning and problem-solving abilities, facilitating effective decision-making and critical thinking in modern tool usage.
CO7: Analyzing minterms, maxterms, disjunctive normal form (DNF), and conjunctive normal form (CNF) is essential in modern tool usage for Boolean algebra simplification and optimization, facilitating efficient digital circuit design and logical expression representation.

PO6: Individual and Team work

CO1: Expressing logic sentences in terms of predicates, quantifiers, and evaluating Boolean functions using Boolean algebra fosters precise communication and efficient problem-solving, promoting both individual cognitive clarity and collaborative teamwork in logical reasoning and decision-making contexts.

CO5: Building and evaluating logical arguments using propositional logic enhances both individual and team problem-solving skills by fostering precise reasoning and collaborative decision-making based on a systematic and structured approach.
CO6: Applying logic concepts in computer science enhances problem-solving by systematically analyzing and structuring information, fostering precision and collaboration in both individual and team work.
CO7: Analyzing minterms, maxterms, disjunctive normal form (DNF), and conjunctive normal form (CNF) facilitates individual and team work by providing a systematic approach to Boolean algebra, aiding in logical circuit design, optimization, and simplification for efficient collaboration in digital system development.

PO7: Innovation, employability and Entrepreneurial skills
CO1: Expressing logic sentences in terms of predicates, quantifiers, and evaluating Boolean functions using Boolean algebra fosters critical thinking and problem-solving abilities, essential for innovation, employability, and entrepreneurial skills by developing a strong foundation in formal reasoning and computational decision-making.
CO2: Transitive closure in graphs enhances innovation, employability, and entrepreneurial skills by identifying and facilitating the exploration of indirect relationships, fostering a holistic understanding of interconnected concepts and opportunities within the dynamic landscape of innovation and entrepreneurship.

CO4: Applying rules of inference in innovation, employability, and entrepreneurial skills enhances logical reasoning and problem-solving, fostering critical thinking essential for effective decision-making and creative problem-solving in dynamic professional environments.

CO6: Applying logic concepts in computer science enhances innovation by fostering critical thinking, ensures employability through problem-solving proficiency, and cultivates entrepreneurial skills by enabling the development of robust and efficient solutions.

CO7: Analyzing minterms, maxterms, disjunctive normal form, and conjunctive normal form enhances innovation and entrepreneurial skills by fostering logical thinking, problem-solving, and structured representation, thereby empowering individuals with employability skills crucial for navigating complex business challenges.

Mapping of Program Outcomes with Course Outcomes

Class: F.Y.B.Sc.(Computer Science). (Sem II)
Subject: Mathematics
Course: Calculus **Course Code:** CSMT1202

A) Course Objectives:

1. To introduce concepts of limit, continuity, differentiation.
2. To understand the behaviour of functions increasing, decreasing, concave up, concave down ,which is crucial in many practical situation.
3. Introduce left-hand and right-hand derivatives.
4. Students will solve first-order linear, separable, exact differential equations.
5. Students will use Taylor's and Maclaurin's series for problem-solving in computer science.
6. Analyze their convergence and divergence properties.
7. Explore applications of differential equations in computer science scenarios.

B) Course Outcome:

1. Understanding of limit concepts and functions properties and their applications.
2. Apply Rolle's Theorem to solve problems.
3. Apply Leibnitz's Theorem to compute derivatives.
4. Students will apply the concept of continuity through examples.
5. Students will comprehend the relationship between continuity and differentiability.
6. Students will interpret Rolle's Theorem geometrically.
7. Students will analyze the convergence and divergence properties of these series.

Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1		1					1
CO 2						1	2
CO 3			2			1	1
CO 4	3		1			2	
CO 5		2				3	
CO 6			3				3
CO 7		3	3			2	

Justification for the mapping

PO1: Computer Knowledge

CO4: Applying continuity in Computer Knowledge fosters seamless adaptation to evolving technologies, ensuring students develop the ability to sustain and advance their skills in a rapidly changing digital landscape.

PO2: Design / Development of solution

CO1: A thorough grasp of limit concepts and function properties is essential for designing and developing effective solutions by providing a foundational understanding of continuity, convergence, and behavior essential for accurate modeling and optimization in various applications.
CO5: Understanding the relationship between continuity and differentiability is essential for designing and developing solutions that require smooth and well-behaved mathematical behavior, ensuring the reliability and effectiveness of the solution.
CO7: Analyzing convergence and divergence properties ensures the robustness and reliability of the solution by verifying the mathematical integrity of the series involved.

PO3: Modern tool usage

CO3:Applying Leibniz's Theorem enhances precision in computing derivatives, facilitating advanced mathematical modeling and analysis in modern computational tools.
CO4: Applying the concept of continuity in modern tool usage enhances students' problem-solving skills and adaptability by fostering a seamless and uninterrupted integration of technological advancements.
CO6: Modern tool usage enhances students' understanding of Rolle's Theorem by providing dynamic visualizations and interactive simulations that facilitate geometric interpretation.
CO7: Students will employ modern tools to assess the convergence and divergence properties of series, enhancing mathematical analysis through computational methods.

PO6: Individual and Team work

CO2: Applying Rolle's Theorem in individual and team work facilitates identifying critical points where performance or collaboration issues exist, enabling targeted interventions to optimize efficiency and enhance overall outcomes.

CO3: Applying Leibniz's Theorem enhances both individual and team efficiency by providing a systematic approach to compute derivatives of higher order, fostering a collaborative problem-solving environment.
CO4: Applying continuity in individual and team work fosters seamless progress, ensuring a fluid integration of efforts and ideas for enhanced collaboration and achievement.
CO5: Understanding the interplay between continuity and differentiability is crucial for students, fostering both individual problem-solving skills and collaborative teamwork by cultivating a foundational grasp of fundamental mathematical concepts.
CO7: Analyzing series convergence and divergence enhances students' mathematical reasoning and teamwork skills by fostering critical thinking, collaborative problem-solving, and a deeper understanding of mathematical concepts.

PO7: Innovation, employability and Entrepreneurial skills

CO1: Understanding limit concepts and function properties is essential for innovation as it forms the foundational knowledge necessary for developing advanced mathematical models, critical for problem-solving and creative solutions. Additionally, these skills enhance employability by fostering analytical thinking and contribute to entrepreneurial success by enabling the identification of opportunities and optimization strategies in various fields.

CO2: Applying Rolle's Theorem in the context of innovation, employability, and entrepreneurial skills enables identifying critical points where the rate of change is zero, facilitating strategic interventions to optimize and enhance key aspects, fostering a more dynamic and effective ecosystem.

CO3: Applying Leibnitz's Theorem enhances innovation, employability, and entrepreneurial skills by providing a systematic approach to calculate derivatives, fostering analytical problem-solving essential for dynamic and rapidly evolving professional environments.

CO6: Interpreting Rolle's Theorem geometrically fosters innovative thinking by linking abstract mathematical concepts with real-world applications, enhancing employability through problem-solving skills and nurturing entrepreneurial mindset through the application of mathematical principles in practical scenarios.

Mapping of Program Outcomes with Course Outcomes

Class: F.Y.B.Sc.(Computer Science). (Sem II)
Subject: Mathematics
Course: Mathematics Practical based on CSMT1201 & CSMT1202 **Course Code:** CSMT1203

A) Course Objectives:

1. Problem solving ability and understanding of applications of Discrete mathematics
2. Analyze equivalence relations, partial orderings, and their applications.
3. Understand logical equivalences involving quantifiers.
4. Apply Boolean algebra to solve problems.
5. Problem solving and understanding of concepts in Calculus such as limit, continuity and differentiation
6. Understand Taylor's and Maclaurin's series with Lagrange's and Cauchy's forms of remainders.
7. Apply Maxima software for solving problems in calculus.

B) Course Outcome:

1. Students will classify different types of relations and analyze their properties.
2. Students will apply propositional equivalences and predicate quantifiers.
3. Students will analyze linear recurrence relations with constant coefficients.
4. Students will analyze properties of continuous functions on closed intervals.
5. Students will understand the relationship between differentiability and continuity.
6. Students will apply Leibnitz's Theorem and explore applications of successive differentiation.
7. Students will utilize Maxima for symbolic computations in calculus.

Weightage: 1= weak or low relation, 2= moderate or partial relation, 3= strong or direct relation

	Programme Outcomes (POs)						
Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7
CO 1	3	1				2	
CO 2			2				1
CO 3			2			1	
CO 4	3						3
CO 5			3			3	
CO 6	1						1
CO 7			3			3	2

Justification for the mapping

PO1: Computer Knowledge

CO1: Understanding and classifying relations is essential in computer knowledge to optimize database design and query efficiency, ensuring effective management and retrieval of interconnected data.

CO4: Understanding properties of continuous functions on closed intervals is crucial for optimizing algorithms and ensuring the stability and reliability of numerical methods used in computer science applications, such as optimization, machine learning, and numerical simulations.
CO6: Applying Leibnitz's Theorem and exploring successive differentiation enhances students' computational skills and analytical thinking, fostering a deeper understanding of mathematical foundations crucial for algorithmic development and optimization in computer science.

PO2: Design / Development of solution

CO1: Understanding and categorizing relations is crucial for designing and developing effective solutions as it forms the foundation for optimizing data structures and establishing logical connections within the system.

PO3: Modern tool usage

CO2: Application of propositional equivalences and predicate quantifiers enhances students' proficiency in modern tools by fostering logical reasoning and problem-solving skills essential for efficient tool utilization.
CO3: Analyzing linear recurrence relations with constant coefficients equips students with essential skills for leveraging modern tools in mathematical modeling, algorithm design, and optimization, fostering a deeper understanding of computational concepts and problem-solving in various disciplines.
CO5: Understanding the relationship between differentiability and continuity is essential for employing advanced mathematical tools in various disciplines, enabling precise analysis and optimization in modern applications such as machine learning, physics, and engineering.
CO7: Students will utilize Maxima for symbolic computations in calculus to enhance their understanding of mathematical concepts through hands-on application, fostering proficiency in modern computational tools for problem-solving in calculus.

PO6: Individual and Team work

CO1: Develops critical thinking and collaborative skills by engaging students in the classification and analysis of various relations, fostering both individual comprehension and teamwork in exploring mathematical properties.

CO3: Analyzing linear recurrence relations with constant coefficients enhances students' problem-solving skills, fosters teamwork, and deepens their understanding of mathematical structures crucial in various disciplines, promoting both individual and collaborative learning.

CO5: Understanding the interplay between differentiability and continuity is crucial for both individual problem-solving and collaborative team efforts in mathematical contexts, fostering a comprehensive grasp of functions and their behavior.
CO7: Maxima facilitates enhanced learning in calculus through its robust symbolic computation capabilities, empowering students to seamlessly apply mathematical concepts in both individual and collaborative settings, fostering a deeper understanding of calculus principles.

PO7: Innovation, employability and Entrepreneurial skills

CO2: Application of propositional equivalences and predicate quantifiers enhances students' logical reasoning and problem-solving abilities, fostering critical thinking essential for innovation, employability, and entrepreneurial success.

CO4: Analyzing properties of continuous functions on closed intervals fosters innovation by cultivating critical thinking and problem-solving abilities, enhances employability through a strong foundation in mathematical reasoning, and nurtures entrepreneurial skills by fostering a deep understanding of foundational concepts crucial for various industries.

CO6: Application of Leibnitz's Theorem and successive differentiation fosters innovation by enhancing problem-solving skills, cultivates employability through advanced mathematical techniques, and nurtures entrepreneurial skills by promoting critical thinking in diverse applications.

CO7: Utilizing Maxima for symbolic computations in calculus enhances students' innovation, employability, and entrepreneurial skills by fostering a deeper understanding of mathematical concepts, promoting problem-solving abilities, and providing practical tools for real-world applications in various fields.