Fergusson College (Autonomous)
Pune

Learning Outcomes-Based Curriculum
for

M.Sc. – I

Industrial Mathematics

with Computer Applications (IMCA)

With effect from June 2019

Programme Structure

Semester	Course	Course Title	Course	No. of
I	MTS4101	Real Analysis	T-Core	4
	MTS4102	Applied Linear Algebra	T-Core	4
	MTS4103	Discrete Mathematics I	T-Core	4
	MTS4104	Software Engineering	T-Core	4
	MTS4105	Experiential Training on C-Programming	P-Core	4
II	MTS4201	Advanced Calculus	T-Core	4
	MTS4202	Abstract Algebra	T-Core	4
	MTS4203	Discrete Mathematics II	T-Core	4
	MTS4204	Probability and Statistics	T-Core	4
	MTS4205	Experiential Training on C++	P-Core	4
	MTS4206	Experiential Training on DBMS	P-Core	4
III	MTS5301	Digital Image Processing	D Elect-1	4
	MTS5302	Statistical Inference	D Elect-2	4
	MTS5303	Complex Analysis	D Elect-3	4
	MTS5304	Financial Mathematics	D Elect-4	4
	MTS5305	Coding Theory	D Elect-5	4
	MTS5306	Operating Systems	T-Core	4
	MTS5307	Computer Networks	T-Core	4
	MTS5308	Experiential Training on Data Structures using C	P-Core	4
Note: Students need to opt any THREE courses from MTS5301 to MTS5305				
IV	MTS5401	DAA	D Elect-1	4
	MTS5402	Cryptography	D Elect-2	4
	MTS5403	Applied Geometry from Computer Graphics	D Elect-3	4
	MTS5404	Dynamical Systems	D Elect-4	4
	MTS5405	Machine Learning with AI	D Elect-5	4
	MTS5406	Theoretical Computer Science	D Elect-6	4
	MTS5407	UNIX Internals	D Elect-7	4
	MTS5408	Experiential Training on Java Programming	P-Core	4
	MTS5409	Experiential Training on R + Python	P Elect-1	4
	MTS5410	Experiential Training on Web UI and UX	P Elect-2	4
Note: Students need to opt **any THREE** courses from MTS5401 to MTS5405, **any ONE** from MTS5406 and MTS5407 and **any ONE** from MTS5409 and MTS5410.				

V	MTS5501	Numerical Analysis	D Elect-1	4
	MTS5502	Optimization Techniques	D Elect-2	4
	MTS5503	Compiler Construction	D Elect-3	4
	MTS5504	Data Mining	D Elect-4	4
	MTS5505	Introduction to UML and Design Patterns	D Elect-5	4
	MTS5506	Mobile Application Development	D Elect-6	4
	MTS5507	Experiential Training on Project Implementation	Project-1	4
Note: Students need to opt **any ONE** from MTS5501 and MTS5502 and **any THREE** from MTS5503 to MTS5506.				
VI	MTS5601	Industrial Training	Industrial Training	8

M.Sc. (Industrial Mathematics with Computer Applications)
Extra Credits

Semester	**Name of the paper**	**Course Code**	**Title of the Course**	**No. of Credits**
I	Extra Credit Course	MTS4121	Human Rights - I	1
	Extra Credit Course	MTS4122	Introduction to Cyber Security - I	1
	Extra Credit Course	MTS4123	Skill Development - I (LATEX)	1
II	Extra Credit Course	MTS4221	Human Rights - II	1
	Extra Credit Course	MTS4222	Introduction to Cyber Security - II	1
	Extra Credit Course	MTS4223	Skill Development - II (SCILAB)	1
III	Extra Credit Course	MTS4321	Introduction to Cyber Security - III	1
	Extra Credit Course	MTS4322	Emerging Trends - I	1
IV	Extra Credit Course	MTS5321	Introduction to Cyber Security – IV	1
	Extra Credit Course	MTS5322	Emerging Trends - II	1

Program Outcomes (POs) for M.Sc. Programme	
PO1	**Disciplinary Knowledge:** Demonstrate comprehensive knowledge of the discipline that form a part of an postgraduate programme. Execute strong theoretical and practical understanding generated from the specific programme in the area of work.
PO2	**Critical Thinking and Problem solving:** Exhibit the skill of critical thinking and understand scientific texts and place scientific statements and themes in contexts and also evaluate them in terms of generic conventions. Identify the problem by observing the situation closely, take actions and apply lateral thinking and analytical skills to design the solutions.
PO3	**Social competence:** Exhibit thoughts and ideas effectively in writing and orally; communicate with others using appropriate media, build effective interactive and presenting skills to meet global competencies. Elicit views of others, present complex information in a clear and concise and help reach conclusion in group settings.
PO4	**Research-related skills and Scientific temper:** Infer scientific literature, build sense of enquiry and able to formulate, test, analyse, interpret and establish hypothesis and research questions; and to identify and consult relevant sources to find answers. Plan and write a research paper/project while emphasizing on academics and research ethics, scientific conduct and creating awareness about intellectual property rights and issues of plagiarism.
PO5	**Trans-disciplinary knowledge**: Create new conceptual, theoretical and methodological understanding that integrates and transcends beyond discipline-specific approaches to address a common problem.
PO6	**Personal and professional competence**: Performindependently and also collaboratively as a part of team to meet defined objectives and carry out work across interdisciplinary fields. Execute interpersonal relationships, self-motivation and adaptability skills and commit to professional ethics.
PO7	**Effective Citizenship and Ethics**: Demonstrate empathetic social concern and equity centred national development, and ability to act with an informed awareness of moral and ethical issues and commit to professional ethics and responsibility.
PO8	**Environment and Sustainability**: Understand the impact of the scientific solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.
PO9	**Self-directed and Life-long learning:** Acquire the ability to engage in independent and life-long learning in the broadest context of socio-technological changes.

Program Specific Outcomes (PSOs) for M. Sc. IMCA	
PSO No.	**Program Specific Outcomes(PSOs)** **Upon completion of this programme the student will be able to**
PSO1	**Academic competence** (i) Understand basic facts about Mathematics -annotations, terminology, geometrical figures, graphical displays and its major subfields (Analysis, Algebra, Applied Mathematics and Statistics). (ii) Design and develop computer programs/computer-based systems in the areas related to algorithms, networking, web design, cloud computing, IoT and data analytics. (iii) Demonstrate unifying structures of Mathematics and the basic constructs of several programming languages and their application areas.
PSO2	**Personal and Professional Competence** (i) Apply mathematical solutions in a variety of contexts related to science, technology, business and industry, and carry out hands- on activities on several programming technologies as well as academic projects. (ii) Analyse the data by selecting and using appropriate mathematical formulae or techniques/programming technology stack in order to draw the relevant conclusion. (iii) Execute the problem solving skills, equip with Mathematical modelling abilities and develop competent professionals who will be able to address challenges in the field of IT at global level.
PSO3	**Research Competence** (i) Apply advanced knowledge on topics in pure Mathematics and impart analytical skills to develop initiatives and come up with innovative ideas for R&D in various fields. (ii) Integrate the knowledge of Computer Science and Mathematics to solve and authenticate real-time data from various fields.
PSO4	**Entrepreneurial and Social competence** (i) Develop analytical skills required to get distinguishing employment opportunities in several fields including IT, Research and Development, teaching field and gain understanding about the ethical issues related to protection of intellectual property - copyrights, trademarks, and patents. (ii) Execute social competence including communication and effective interaction with others, listening, speaking, observational skills and presenting skills.

Course Outcome (COs)		
F.Y. M.Sc. Semester I		
Title of the Course and Course Code	**Real Analysis (MTS4101)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Retrieve basic concepts in Metric Topology.	
CO2	Interpret properties of continuous and differentiable functions.	
CO3	Implement Mean Value Theorems to real life problems and discuss its use.	
CO4	Identify series expansions for general category of continuously differentiable functions for computational needs.	
CO5	Evaluate Riemann Integration to generalise the theory of Integration over Euclidean Spaces.	
CO6	Formulate the Fundamental Theory of Integral Calculus, support the theory with examples and create counter examples wherever needed.	

Unit No.	Title of Unit and Contents
I	**Basic Topology** Finite, Countable and uncountable sets, Metric Spaces, Compact Sets, Perfect Sets, Connected Sets.
II	**Numerical Sequences and Series** Convergent Sequences, Subsequences, Cauchy Sequences, Some special Sequences Series , Series of Non-negative Terms, The number e, The Root and Ratio Tests, Power Series, Summation by parts, Absolute Convergence.
III	**Continuity** Limits of Functions, Continuous Functions, Continuity and Connectedness, Continuity and Compactness, Discontinuities, Monotonic Functions.
IV	**Differentiation** Derivatives and Mean Value Theorems, Taylor’s Theorem, Convex Functions, Cauchy form of the remainder term, Differentiation of Vector Valued Functions.
V	**Riemann Stieltjes Integral** Definition and Existence of the Integral, Properties of the integral, Integration and Differentiation, Integration of Vector Valued Functions, Rectifiable Curves.

Learning Resources

- Walter Rudin, Principles of Mathematical Analysis, McGraw Hill India (3rd edition)
- Ajit Kumar and S. Kumerasan, A First Course on Real Analysis CRC Press

Title of the Course and Course Code	**Applied Linear Algebra (MTS4102)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Describe key concepts associated with Vector spaces.	
CO2	Illustrate various mathematical properties of inner products and solve examples to justify the properties.	
CO3	Apply the concept of orthogonality to find an orthogonal basis using the Gram Schmidt process. Compute Eigen values and Eigenvectors for a given matrix	
CO4	Identify the role of Eigen values and Eigenvectors in Matrix decompositions.	
CO5	Determine the concept of Linear Transformations to solve real life problems.	
CO6	Formulate the concept of matrix decompositions to discuss problems in Engineering and Data Science	

Unit No.	Title of Unit and Contents
I	**Vector Spaces and Basis** Real Vector Spaces**,** Subspaces, Intersection of Subspaces, Sum of Subspaces**,** Span and Linear Independence**,** Basis and Dimensions**,** The Fundamental Matrix Subspaces, Kernel and Image, Superposition Principle, Adjoint Systems, Cokernel and Coimage, Fundamental theorem of Linear Algebra
II	**Inner Products and Norms** Inner Products, Inner Product on Function Spaces, Inequalities: Cauchy Schwarz, Orthogonal Vectors, Triangle Inequality, Norms: Unit Vectors and Matrix Norms, Positive Definite Matrices : Gram Matrices, Completing the Square: Cholesky Factorization
III	**Orthogonality** Concepts of Orthogonal and Orthonormal Basis, Gram Schmidt Process, Orthogonal Projections and Orthogonal Subspaces, Orthogonality of Fundamental Matrix Subspaces and the Fredholm Alternative, Orthogonal Polynomials : Legendre Polynomials
IV	**Linearity** Linear Operators, Space of Linear Functions, Dual Spaces, Composition and Inverses**,** Linear Transformations, Change of Basis**,** Introduction to Affine Transformations and Isometry**,** Adjoints, Positive Definite Linear Functions and Minimization
V	**Eigen values and Eigenvectors** Introduction, Basic Properties and Gerschgorin Circle Theorem Eigenbasis, Diagonalization, Invariant Subspaces, Eigenvalues of Symmetric Matrices, Spectral theorem (Statement only) Introduction to Schur’s Decomposition and Jordan Canonical Form

	Introduction to Singular values, The Pseudo Inverse, Euclidean Matrix Norm, Conditional number and rank, Variance Covariance and Introduction to Principal Components
VI	**Applications** Minimization of Quadratic Forms, Concept of the Closest Point Least Squares, Introduction to Discrete Fourier Transform, Compression and Denoising, Introduction to Haar Wavelets, Modern Wavelets, Haar Scaling Function.

Learning Resources

- Peter Olver and ChehradShakiban, Applied Linear Algebra (Second Edition) Springer Publishing house
- Gilbert Strang, Linear Algebra and its applications (Fourth Edition)
- K.Hoffman and Ray Kunje , Linear Algebra (PHI India Private Ltd)
- David Lay, Linear Algebra with Applications
- M.L.Artin , Algebra (PHI private Limited)
- A.G.Hamilton ,Linear Algebra (Cambridge University House)
- Henry Helson, Linear Algebra (Hindustan Book Agency)
- I.N.Herstein ,Topics in Algebra (Second Edition)

Title of the Course and Course Code	**Discrete Mathematics I (MTS4103)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the propositional equivalences, quantifiers, predicates and different types of proofs	
CO2	Articulate basic concepts of Logic	
CO3	Apply basic graph theory, minimal weighted spanning tree algorithms, graph colouring algorithms	
CO4	Compare different types of graphs and operations on graphs.	
CO5	Determine algorithms of fusion, matching algorithms, shortest path algorithms Formulate	
CO6	Formulate Matrix representation of graphs, shortest paths for various graphs	

Unit No.	Title of Unit and Contents
I	**Logic** Introduction, Proposition, Simple proposition, compound proposition, Truth value, Prepositional Calculus, operators, Conjunction, Disjunction, conditional statement, Biconditional statement, converse, contrapositive and Inverse, Precedence of logical operators, Translating in English sentences into symbolic form and logical implications. **Propositional Equivalences** Introduction, Logical equivalences, Tautology, Contradiction, Logic rules. **Predicates and Quantifiers** Introduction, Universal quantifier, existential quantifier, counter examples, negating quantifiers, translating sentences into logical expressions, nested

	quantifiers, truth value of quantifiers. **Methods of Proof** Theorem, Proof, rules of inference, argument, valid argument, invalid argument, direct method of proof, indirect method of proof, rules of reference for quantified statements.
II	**Graph Theory: [36 Lectures]** **Graph:** Definition, vertex, edge, Terminal vertices, self-loop, parallel edges, incidence, adjacent, degree of vertex, isolated vertex, pendent vertex, null graph, hand shaking lemma, regular graph, bipartite graph, complete graph, complete bipartite graph. **Matrix Representation:** Incidence matrix, adjacency matrix, properties. Subgraph, Isomorphism and examples of isomorphic graphs. **Operations on graphs:** Union, intersection, deletion of vertex, deletion of edge, ring sum, fusion. **Connected graphs:** Walk, paths, circuit, Theorems on connected graphs. **Euler graph:** Definition, examples, Chinese postman problem, Fleury's algorithm, Theorems on Eulerian graphs. **Trees:** Definition, pendent vertex in a tree, distance and centres in a tree, rooted and binary trees, spanning trees and rank nullity, fundamental circuits, fundamental cutset, vertex connectivity, edge connectivity, spanning trees, weighted graphs, Kruskal's algorithms, Prim's algorithm, Breadth first search algorithm, depth first search algorithm, Dijkstra's algorithm, Warshall Floyd algorithm, Theorems on trees. **Directed graphs:** Incident out of a vertex, incident into a vertex, indegree, out degree, isolated vertex, pendent vertex, types of digraphs, arborescence definition. **Networks** Flows and cuts, Max ow and min cut theorem, The Ford and Fulkerson Algorithm **Graph Coloring:** **Vertex Coloring:** K-coloring, K-colorable, Chromatic Number, K-Chromatic. Vertex coloring Algorithm: Simple Sequential Coloring, Largest–First Sequential Algorithm (Welsh and Powell) Smallest–Last Sequential Algorithm. **Edge Coloring:** Definition and Concept Only. **Planar Graphs:** Introduction Kuratowski's two graphs (K5, K3) Euler's theorem, Examples based on Euler's theorem. **Matching and Factors:** Matching in bipartite graphs, maximum matchings, Hall`s matching conditions, Min-Matching in bipartite graphs, sets, applications and algorithms, maximum bipartite matching, weighted bipartite matching, Tutte`s 1 factor theorem, factors of graphs.

Learning Resources

- Kenneth H. Rosen, Discrete Mathematics and its Applications (TATA McGraw – HILL), Edition 6
- Kolman, Busby, Ross and Rehman, Discrete Mathematical Structures, Pearson Edition Sixth Edition
- John Clark and Derek Allan Holton, A first look at Graph Theory
- N. Deo, Graph theory with Applications to Computer Science and Engineering, PHI

- publication.
- Douglas B. West, Introduction to Graph Theory, Pearson Education, Second Edition
- Purna Chandra Biswal, Discrete Mathematics and Graph Theory, Fourth Edition (PHI.).
- Alan Tucker, Applied Combinatorics, John Willey, Fourth Edition.

Title of the Course and Course Code	**Software Engineering (MTS4104)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Recall facts about software engineering.	
CO2	Articulate the key concepts, characteristics, objectives and applications of software engineering.	
CO3	Apply the concepts of software requirements in the software requirement engineering process ,design techniques, testing strategies.	
CO4	Analyze different life cycle models, structures through different tools such as data flow diagrams, structure charts, decision tables and decision trees based on their strengths and weaknesses	
CO5	Discriminate different life cycle models, Test and maintenance techniques	
CO6	Build the foundation for requirement analysis with the use of software requirement engineering concepts	

Unit No.	Title of Unit and Contents
I	**Introduction to software engineering** What is software engineering, software engineering principles, Software characteristics, applications, Objectives of software engineering, Phases of software engineering
II	**Software Processes and Life Cycle models** Software process, project and product, process assessment, Software Process capability maturity model: CMM Model. Life cycle models: Waterfall model, Incremental model, spiral model, advantages and disadvantages, Prototyping Model, Object-oriented model, Agile model, Extreme programming (Latest models can be discussed), advantages and disadvantages.
III	**Software requirements and Software Requirement engineering process** Functional, non-functional requirements, User requirement, System requirements, Software requirements documentation, Feasibility studies, Requirements elicitation and analysis, requirement validation, software prototyping, requirement management.
IV	**Software Reliability** Software Reliability; Software Reliability Metrics; Programming for Reliability; Software Reuse.
V	**Software design** Basics of software design, Data design, Architectural design, component level design and user interface design ,Fundamental design concepts-module

	and modularization, Design techniques
VI	**Structure analysis and tools** Data flow diagrams, Structure charts, decision tables, decision trees
VII	**Software maintenance** Software re-engineering, Change management, configuration management, maintenance tools and techniques.
VIII	**Software testing strategies:** A strategic approach to software testing, test strategies for convention software, Black-box and white box testing, validation and system testing, and debugging.

Learning Resources

- Jessica Keyes. *Software Engineering Handbook*. Auerbach Publications (CRC Press), 2003.
  Contains complete examples of various SE documents.
- Roger S. Pressman. *Software Engineering: A Practitioner's Approach* (Sixth Edition, International Edition). McGraw-Hill, 2005.
- Ian Sommerville. *Software Engineering* (Seventh Edition). Addison-Wesley, 2004.
- Hans van Vliet. *Software Engineering: Principles and Practice* (Second Edition). Wiley, 1999.

Title of the Course and Course Code	**Experiential Training Course on C Programming (MTS4105)**	**Number of Credits : 04**

Course Outcomes (COs) **On completion of the course, the students will be able to:**	
CO1	Recall the basic concepts of the C programming language.
CO2	Discuss basic concepts of computers, algorithms and algorithmic thinking.
CO3	Apply stack for implementation of function calls and parameter passing mechanisms.
CO4	Analyze and compare usage of arrays, strings, structures and files.
CO5	Determine the solution for a given problem by distinguishing various memory allocation methods.
CO6	Write C programs to validate the specifications.

Unit No.	Title of Unit and Contents
I	**Introduction to Programming** Program and Programming, Programming Languages, Types of Software, Operating Systems, Basic Linux Commands and vi Editor, Compiler, Interpreter, Loader and Linker
II	**Basics of C** History and Features of C, Importance of C, Backslash Characters, Character set, Constants, Format Specifiers, Identifiers, Keywords, Variables, Data Types, Comments, const Qualifier, The Structure of a C Program, Building an Executable Version of a C Program, Debugging a C Program, Programming Examples

III	**Applications of C Programming** Demonstration of an application developed using C Note: This unit will not be considered for an assessment of students.
IV	**Control Statements** Decision Making Statements: if, if-else, switch Loop Control Structures: while, do. while, for Keywords- break and continue, exit() Function, return Statement, Programming Examples
V	**Operators and Expressions** Arithmetic Operators, Increment and Decrement Operators, Relational Operators , Logical Operators, Bitwise Operators, Assignment Operators, Conditional Operator, size of Operator, Comma Operator, Type Casting Operator, Other Operators, Precedence and Order of Evaluation, Programming Examples
VI	**Input and Output** Unformatted I/O, Character I/O, String I/O, Formatted I/O, Programming Examples
VII	**Functions** Concept, Usage of a Function, Advantages, Function Prototype, Function example, Types of Function, Call by Value and Call by Address, Recursion, Library Functions, Local variable, Global Variable, Storage classes (automatic, static, register, external), Programming Examples
VIII	**Array** Array Declaration, Initialization, Types of Array (1-D, 2-D and Multidimensional), Passing Arrays to Functions, Programming Examples
IX	**Pointers** Pointer Declaration and Initialization, Dereferencing Pointers, void Pointer, Pointer Arithmetic, Pointer to Pointer, Arrays and Pointers, Functions and Pointers, Passing Pointers to Function, Function Returning Pointer, Pointer to Function, Dynamic Memory Allocation, Programming Examples
X	**String Handling** Declaration and Initialization, Reading and Writing Strings, Standard String Library Functions, Array of Pointers to String, Command Line Arguments, Programming Examples
XI	**Structures and Unions** Overview of Structures, Defining and Using a Structure, typedef Keyword, Nested Structures, Passing Structure to Function, Structure and Pointer, Union, Difference between Structure and Union, Programming Examples
XII	**Pre-Processor Directives** Pre-Processor Directives, #define Macro, Conditional Compilation, Pre-defined Macros, #include and Header Files, Programming Examples
XIII	**File Handling** What is a Stream? Opening and Closing of Files, File Opening Modes, Writing and Reading in Text Format, Writing and Reading in Binary Format, Programming Examples

Learning Resources

Include Reference Books/ e-resources / journals/any other learning material

1. Kernighan Brian W., Ritchie Dennis M., The C Programming Language, PHI Learning Pvt. Ltd., 2nd Edition, 2010
2. Schildt Herbert, C: The Complete Reference, Tata McGraw Hill, 4th Edition, 2006
3. KanetkarYashavant, Pointers in C, BPB Publications, 4th Edition, 2013
4. KanetkarYashavant, Test your C Skills, BPB Publications, Rev. Edition, 2008

F.Y. M.Sc. Semester II		
Title of the Course and Course Code	**Advanced Calculus (MTS4201)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Describe basic concepts related to sequences and series of functions.	
CO2	Discuss various properties related to sequence and series of functions and illustrate it with suitable examples.	
CO3	Demonstrate the concept of Integral of a k-form considered over a boundary and generalise it to arbitrary spaces. Apply the calculus concepts and solve the problems.	
CO4	Explain various properties of elementary functions to solve problems in science and engineering applications.	
CO5	Evaluate derivatives of scalar, vector valued functions to determine maxima, minima and saddle points.	
CO6	Develop the Stoke Theorem to structure the concept of Integration to be stated by a single formula.	

Unit No.	Title of Unit and Contents
I	**Sequences and Series of Functions** Uniform Convergence, Uniform Convergence and Continuity, Uniform Convergence and Differentiation, Uniform convergence and Integration, Equicontinuous Families of Functions, The Stone Weierstrass Theorem
II	**Special Functions** Power Series, The Exponential and Logarithmic Functions, The Trigonometric Functions, Fourier Series, Gamma Function.
III	**Functions of Several Variables** Linear Transformations, Differentiation, The Contraction Principle The Inverse Function Theorem, The Implicit Function Theorem, The Rank Theorem, Determinants, Derivatives of higher Order Differentiation of Integrals
IV	**Integration of Differential Forms** Integration, Primitive Mappings, Partitions of Unity, Change of Variables, Differential Forms, Simplexes and Chains, Stoke's Theorem, Closed and Exact Forms, Vector Analysis.

Learning Resources

- Walter Rudin, Principles of Mathematical Analysis, McGraw- Hill India (3rd edition)
- Michael Spivak , Calculus of Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus, Benjamin Cummings
- James Munkres, Analysis of Manifolds, Mathematical Association of America

Title of the Course and Course Code	**Abstract Algebra (MTS4202)**	**Number of Credits : 04**
Course Outcomes (COs) **On completion of the course, the students will be able to:**		
CO1	Define basic concepts of Group theory with focus on axiomatic theory and key concepts associated with graphs. State	
CO2	Articulate the fundamental concepts of abstract algebra such as groups and rings and their role in modern Mathematics and applied contexts.	
CO3	Demonstrate capacity for mathematical reasoning through analyzing, proving, and explaining concepts from abstract algebra. Describe the structure of certain finite groups using Sylow's theorems.	
CO4	Compare different types of groups and rings. Explain the notion of rings, ideals.	
CO5	Justify theorems based on group theory, ring theory and articulate problem-solving techniques based on them.	
CO6	Construct the structure of finite fields.	

Unit No.	Title of Unit and Contents
I	**Groups** Definitions and Examples, Simple properties of Groups based on axioms, Order of an element, properties and example. Subgroups: Definition and examples, necessary and sufficient condition for non-empty subset to be a subgroup, properties of subgroup, cyclic subgroups, Definition and examples of cyclic subgroup, properties, counting principle, cosets- examples and properties, Lagrange's theorem and its corollaries. Permutation groups: Definition and Examples, Permutation as composition of function, Definition of $\mathbf{S_n}$and discussion of S_3 and S_4 in detail. Cycles, Transpositions, Every Permutation is a product of disjoint cycles, Even and odd permutations, order of a permutation, Alternating group $\mathbf{A_n}$. Homomorphism and Isomorphism: Definitions and Examples, Simple Properties Isomorphism - Definition and Examples Fundamental theorem of homomorphism and its applications, Cayley's theorem Normal Subgroups: Definition and Examples, Properties of Normal Subgroups, Simple Groups, $\mathbf{A_n}$is simple for n=5, Factor Group, Definition and Examples, Properties of Factor groups. Sylow's theorems: Class Equations, Conjugate of an Element-Definition and Examples, Conjugacy relation is and equivalence relation, Conjugacy Class Normalizer, Centralizer, Centre of a group, Class equation, a belongs to Z(G) if and only if N(a) = G, Centre of a p-group is nontrivial, every group of

	order p-square is abelian. Cauchy's theorem, Sylow's theorems (without proofs) - only problems.
II	**Rings [18 Lectures]** Definitions and examples, simple properties of rings, Commutative rings, ring with unity, integral domain, field, skew field, definitions examples and interrelationship between them. Subring: Definition, Examples, Properties. Characteristic of an integral domain. Ideals and Factor Rings: Definitions & Examples, Properties of ideals, Prime Ideals, Maximal Ideals, Quotient rings Homomorphism and Isomorphism of rings: Definition and examples, properties of ring homomorphisms, fundamental theorem of ring homomorphisms and its applications. Euclidean rings: Polynomial rings F[X] over a field F, F[X] is Euclidean ring, irreducible polynomial over a field, polynomials over a field of rationals, Gauss lemma and Eisenstein's criterion for irreducibility, Construction of finite fields.

Learning Resources

- Contemporary Abstract Algebra by Joseph Gallian (Fourth Edition, Narosa Publication)
- J.B. Fraleigh, Abstract Algebra, 7th edition
- I.S. Luthar and I.B.S. Passi, Algebra (Volume 1) Groups (Narosa Publishing House)
- I.N. Herstein, Topics in Algebra (Wiley-Eastern Ltd)
- M. Artin, Algebra (Prentice Hall)
- N.S. Gopala Krishnan, University Algebra (Wiley-Eastern Ltd)
- David S. Dummit and Richard M. Foote, Abstract Algebra (Wiley-Eastern Ltd), Second edition.

Title of the Course and Course Code	**Discrete Mathematics II (MTS4203)**	**Number of Credits : 04**
Course Outcome (COs) On completion of the course, the students will be able to:		
CO1	State counting principles.	
CO2	Articulate counting principles, recurrence relations.	
CO3	Apply fundamental notions of Lattice theory ,properties of Lattices and basics of Boolean axioms	
CO4	Explain fundamental operations on Boolean expressions.	
CO5	Conclude application areas of Discrete Mathematics	
CO6	Develop an ability to solve individually and creatively advanced problems connected with its applications to Mathematics.	

Unit. No.	Title of Unit and Contents
I	**Logic** Introduction, Proposition, Simple proposition, compound proposition, Truth value, Prepositional Calculus, operators, Conjunction, Disjunction, conditional statement, Biconditional statement, converse, contrapositive and Inverse, Precedence of logical operators, Translating in English sentences

	into symbolic form and logical implications. **Propositional Equivalences** Introduction, Logical equivalences, Tautology, Contradiction, Logic rules. **Predicates and Quantifiers** Introduction, Universal quantifier, existential quantifier, counter examples, negating quantifiers, translating sentences into logical expressions, nested quantifiers, truth value of quantifiers. **Methods of Proof** Theorem, Proof, rules of inference, argument, valid argument, invalid argument, direct method of proof, indirect method of proof, rules of reference for quantified statements.
II	**Graph Theory: [36 Lectures]** **Graph:** Definition, vertex, edge, Terminal vertices, self-loop, parallel edges, incidence, adjacent, degree of vertex, isolated vertex, pendent vertex, null graph, hand shaking lemma, regular graph, bipartite graph, complete graph, complete bipartite graph. **Matrix Representation:** Incidence matrix, adjacency matrix, properties. Subgraph, Isomorphism and examples of isomorphic graphs. **Operations on graphs:** Union, intersection, deletion of vertex, deletion of edge, ring sum, fusion. **Connected graphs:** Walk, paths, circuit, Theorems on connected graphs. **Euler graph:** Definition, examples, Chinese postman problem, Fleury's algorithm, Theorems on Eulerian graphs. **Trees:** Definition, pendent vertex in a tree, distance and centres in a tree, rooted and binary trees, spanning trees and rank nullity, fundamental circuits, fundamental cutset, vertex connectivity, edge connectivity, spanning trees, weighted graphs, Kruskal's algorithms, Prim's algorithm, Breadth first search algorithm, depth first search algorithm, Dijkstra's algorithm, Warshall Floyd algorithm, Theorems on trees. **Directed graphs:** Incident out of a vertex, incident into a vertex, indegree, out degree, isolated vertex, pendent vertex, types of digraphs, arborescence definition.

Learning Resources

- Kenneth H. Rosen, Discrete Mathematics and its Applications (TATA McGraw – HILL), Edition 6
- Kolman, Busby, Ross and Rehman, Discrete Mathematical Structures, Pearson Edition Sixth Edition
- John Clark and Derek Allan Holton, a first look at Graph Theory
- N. Deo, Graph theory with Applications to Computer Science and Engineering, PHI publication.
- Douglas B. West, Introduction to Graph Theory, Pearson Education, Second Edition
- Purna Chandra Biswal, Discrete Mathematics and Graph Theory, Fourth Edition (PHI.).
- Alan Tucker, Applied Combinatorics, John Willey, Fourth Edition.

Title of the Course and Course Code	**Probability and Statistics (MTS4204)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	State basic concepts related to Probability Theory and solve examples.	
CO2	Discuss various properties related to mean and variance.	
CO3	Apply the concept of discrete and continuous distributions to a given model or a given data.	
CO4	Analyze the concept of Correlation and Regression for a given data.	
CO5	Determine the statistical significance of null hypothesis with inference based tests to various problems.	
CO6	Create a Regression Model on a given random dataset to find out the statistical significance of various models and verify the assumptions.	

Unit No.	Title of Unit and Contents
I	**Introduction to Probability** Classical Definition of Probability, Mutually Exclusive and Independent Events, Inclusion Exclusion Principle, Conditional Probability, Bayes Theorem and related problems
II	**Random Variables and Distribution Functions** Introduction, Distribution Functions, Discrete Random Variable, Examples, Concept of Probability Mass Functions, Continuous Random Variables, Examples and Concept of Probability Density Functions. Introduction to Joint Probability Mass Functions and Marginal Distributions, Expected Value and Variance
III	**Introduction to Distributions** Introduction to Binomial, Poison, Geometric, Continuous Uniform, Normal, Exponential and problems related to these distributions.
IV	**Important Results and Theorem** Discussion on the Random Walk problem, Statements on various versions of Central Limit theorem. Statement of Chebychev's Inequality and Weak Law of Large Numbers.
V	**Correlation and Regression** Introduction, Scatter Diagrams, Karl Pearson Correlation Coefficient, properties and problems related to it, Introduction to Simple Linear Regression Model, Regression Equations, Examples, Assumptions of the Model, Introduction to Multiple Linear Regression Model, Regression Equations, Assumptions of the Model and Fitting of the Models.
VI	**Testing of Hypothesis** Introduction to Sampling, Basic Terms of hypothesis testing, Type I, Type II errors, Large Sample tests based on Means and Proportions, Introduction to chi-square, t and F distributions, Small sample tests based on t for means, difference of means, Paired t test, chi-square test for Independence of attributes, goodness of Fit, Introduction to Modeling of an One way ANNOVA and a problem related to it.

Learning Resources:

- Sheldon Ross, Probability and Statistics for Engineers and Scientists, Academic Press (Fourth Edition)
- George Casella, Roger Lee Berger, Statistical Inference, Wordpress
- S.C. Gupta , Fundamentals of Statistics, Himalaya Publishing House

Title of the Course and Course Code	**Experiential Training Course on C++ (MTS4205)**	**Number of Credits : 04**
Course Outcome (COs) **On completion of the course, the students will be able to:**		
CO1	Describe the features of Object-Oriented Programming using C++.	
CO2	Explain containment and inheritance to promote code reuse in C++.	
CO3	Implement file handling operations using C++ programming.	
CO4	Analyze the strengths of C++ programming.	
CO5	Test and validate C++ applications using exception handling mechanism.	
CO6	Write object-oriented applications using C++.	

Unit No.	Title of Unit and Contents
I	**Introduction to C++** History, Features of C++, Structure of C++ program, Variables, Data Types, Keywords, Operators, Namespaces, using Keyword, I/O Stream, References in C++, C vs C++, Programming Examples
II	**Control Statements** Decision Making Statements: if, if-else, switch Loop Control Structures: while, do. While, for Keywords- break and continue, Comments, Programming Examples
III	**Functions** Concept, Usage of a Function, Types of Function, Call by Value, Call by Reference and Call by Address, References vs Pointers, Return by Reference, Inline Function, Default Arguments Concept, Recursion, Programming Examples
IV	**Arrays** Arrays, Passing Array to Function, Multidimensional Arrays, Programming Examples
V	**Strings** Concept, Operations on Strings, Standard Library String Functions, Programming Examples
VI	**Class and Objects** OOPs concepts: Encapsulation, Inheritance, Polymorphism, Abstraction Object, Class, Constructor, Types of Constructor (Default, Parameterised, Copy), Destructor, Virtual Destructor, this Pointer, static Members (Fields & Member Functions), Structs, Friend Function, Programming Examples

VII	**Inheritance & Aggregation** Concept, Advantages, Types of Inheritance, Aggregation, Programming Examples
VIII	**Polymorphism** Concept, Function overloading, Operator overloading, Function overriding, Virtual function, Virtual base class, Programming Examples
IX	**Abstract class** Concept, Pure Virtual Function, Interface, Programming Examples
X	**File & Stream** Concept, I/O Manipulators (endl, flush, setfill, setprecision, setw), fstream, ifstream, ofstream, File I/O, Programming Examples
XI	**Exception Handling** Concept, Exception Handling Keywords (try, catch, throw), Advantages, Standard Exception Classes in C++, User-defined Exceptions, Programming Examples
XII	**Templates** Concept, Function Template, Overloading of Function Template, Restrictions on Generic Functions, Class Template, Programming Examples
XIII	**Introduction to STL (Standard Template Library)** Concept, Containers (Stack, Queue, Vector, List), Algorithms (Sorting, Searching), Iterator, Programming Examples

Learning Resources

1. Herbert Schildt, The Complete Reference C++, Tata McGraw Hill, 4th Edition, 2003
2. Herbert Schildt, C++ Programming Cookbook, Tata McGraw Hill, 2008
3. Bjarne Stroustrup, The C++ Programming Language, Pearson, 4th Edition
4. Lafore Robert, Object-Oriented Programming with C++, Pearson, 4th Edition, 2010
5. Kanetkar Yashavant, Let us C++, BPB Publications, 2nd Edition, 2010

Title of the Course and Course Code	**Experiential Training Course on DBMS (MTS4206)**	**Number of Credits : 04**
	Course Outcome (COs) **On completion of the course, the students will be able to:**	
CO1	Describe major components of DBMS.	
CO2	Interpret a problem and to recognize the computing requirements appropriate to its solution	
CO3	Implement appropriate database for computer-based systems according to the user requirements, appropriate syntax to write SQL commands to perform various RDBMS operations.	
CO4	Analyze a problem to found out the computing requirements appropriate to its solution.	
CO5	Discuss the purpose of query processing for optimized solution.	
CO6	Design data requirements of an application with the help of conceptual modelling tools.	

Unit No.	Title of Unit and Contents
I	**Database Management System Concepts**: 1.1 Introduction 1.2 Significance of Database 1.3 Database System Applications 1.4 Data Independence 1.5 Entities and their Attributes 1.6 Relationships and Relationships Types 1.7 E R Diagram 1.8 Data types 1.9 Creating tables (without keys)
II	**An Introduction to RDBMS**: 2.1 Relational Database Management System 2.2 RDBMS Properties 2.3 Maintaining Integrity and Defining Data Integrity 2.5 Integrity Rules and Integrity Constraints 2.6 Relational Integrity Rules 2.7 Creating tables (with keys)
III	**SQL** 3.1 Types of SQL DCL- DML 3.2 Basic queries in SQL Single table 3.3 Deletion- Insertion- and Update in SQL 3.4 Simple queries (with insert, delete, and update)
IV	4.1 Multi table Retrievals 4.2 Nested queries (with foreign key and using multi tables)
V	**Stored Functions** 5.1 Function definition 5.2 How to write function and its execution 5.3 Solving some problems with function
VI	**Stored Procedures** 6.1 Procedure definition 6.2 How to write procedure and its execution 6.3 Solving some problems with procedure
VII	**Cursors** 7.1 Cursor definition 7.2 How to write cursor and its execution 7.3 Solving some problems with cursor
VIII	**Triggers** 8.1 Trigger definition 8.2 How to write trigger and its execution 8.3 Solving some problems with trigger
IX	**Views** 9.1 View definition 9.2 How to write view and its execution 9.3 Solving some problems with view

Learning Resources

1. Henry F. Korth, Abraham Silberschatz, S. Sudarshan Database System Concepts, ISBN:9780071289597, Tata McGraw-Hill Education
2. KorryDouglas, PostgreSQL, , ISBN:9780672327568
3. John Worsley, Joshua Drake Practical PostgreSQL (B/CD), ISBN: 9788173663925 Shroff / O'reilly
4. Joshua D. Drake, John C Worsley Practical Postgresql, O’Reilly
5. Richard Stones, Neil Matthew Beginning Databases with PostgreSQL, From Novice
to Professional, 2nd Edition