M.Sc. (Mathematics and Computing)
Courses and Syllabus

School of Mathematics

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

2022

M.Sc. Mathematics and Computing

Name of Programme: Master of Science (Mathematics and Computing).

Introduction Mathematics and Computing Programme is combination of mathematics with computer science. Covering the major areas in demand today, this programme is of utmost value to the aspiring graduate students with Mathematics background. The course provides students with comprehensive theoretical knowledge and also practical training in computer science and numerical computing. This programme has been introduced due to the need for sophisticated mathematics for modern scientific investigations and technological developments. The curriculum is designed to provide students with in depth theoretical background and practical training in computer science and numerical computing so that a student become competent to take challenges in mathematics at national and international levels.

Nature: Full time/ Part time/ Correspondence: Full Time.

Duration: Two Years (4 Semesters).

Eligibility Criteria and Admission Procedure: The candidates seeking admission to M.Sc. (Mathematics & Computing) must have a Bachelor degree with Mathematics as a major subject. The qualifying degree must be from a recognized University by the University Grants Commission with minimum duration of three years. The candidate must have at least 60% (55% for SC/ST) marks in qualifying degree. Admissions shall be made by merit which will be made by combining the percentages of marks obtained in 10th, 12th and graduation level. The degree marks will be considered up to second year/four semesters.

Number of Seats: 20.

Program Educational Objective: The objectives of the M.Sc. (Mathematics and Computing) program are to

1. create a platform for higher studies and research in mathematics, computing, and inter-disciplinary areas.
2. develop sound analytical and practical knowledge to take up jobs in Software, Finance, Industry, Research, and Teaching.
3. prepare students to qualify various national and international competitive examinations.

Program Outcomes: The successful completion of this program will enable the students to

1. acquire the knowledge and explaining of the pure mathematics covering analysis and algebra and ability to apply this knowledge in other fields.
2. use of applied mathematics courses such as Numerical Analysis, Operations Research, Probability and Statistics; and Mechanics to solve real life problems.
3. join software and IT industry with sound knowledge of programming and mathematics based computing.
4. pursue research as a career in mathematics, computing and inter-disciplinary fields.

M.Sc. Mathematics & Computing : Courses[1]

First Semester

S.No.	Course No.	Course Name	L	T	P	Credits
1.	PMA107	Real Analysis	3	1	0	3.5
2.	PMA108	Algebra I	3	1	0	3.5
3.	PMA109	Ordinary Differential Equations	3	1	0	3.5
4.	PMA110	Mechanics	3	1	0	3.5
5.	PMA111	Computer Programming	3	0	2	4.0
6.	PMC105	Discrete Mathematical Structures	3	1	0	3.5
		Total	18	5	2	21.5

Second Semester

S.No.	Course No.	Course Name	L	T	P	Credits
1.	PMA204	Measure Theory and Integration	3	1	0	3.5
2.	PMA205	Algebra II	3	1	0	3.5
3.	PMC205	Database Management System	3	0	2	4.0
4.	PMA207	Complex Analysis	3	1	0	3.5
5.	PMA208	Numerical Analysis	3	1	2	4.5
6.	PMC212	Data Structure and Algorithms	3	0	4	5.0
		Total	18	4	8	24

Third Semester

S.No.	Course No.	Course Name	L	T	P	Credits
1.	PMA301	Functional Analysis	3	1	0	3.5
2.	PMC206	Operating Systems	3	0	2	4.0
3.	PMA303	Probability and Statistics	3	1	2	4.5
4.	PMA304	Mathematical Programming	3	1	2	4.5
5.	PMC303	Computer Network	3	0	2	4.0
6.		Elective-I	3	0	2	4.0
		Total	18	3	6	24.5

Fourth Semester

S.No.	Course No.	Course Name	L	T	P	Credits
1.	PMA401	Number Theory	3	1	0	3.5
2.	PMA402	Mathematical Methods	3	1	0	3.5
3.		Elective II	3	0	2	4
4.	PMC492	Dissertation				10
		Total	9	2	2	21

Total Credits: 91

[1]Approved in 107th Senate meeting held on June 16, 2022 (Revised).

List of Electives

Elective I [4 Credits]

S.No.	Course No.	Course Name
1.	PMC328	Computer & Network Security
2.	PMC329	Parallel and Distributed Computation
3.	PMC330	Interactive Computer Graphics
4.	PMC331	Machine Learning and Applications

Elective II [4 Credits]

S.No.	Course No.	Course Name
1.	PMC434	Data Analysis and Visualization
2.	PMC435	Principles of Secure Coding
3.	PMC436	GPU Programming
4.	PMC437	Computer Vision: Algorithms and Applications

PMA107 Real Analysis

Credits: 3.5 [3-1-0]

Course Objectives: The aim of this course is to introduce the students real number system and metric spaces. In particular, the notion of completeness, compactness, limit, continuity, differentiability, integrability and uniform continuity.

Contents:

Real Number System and Set Theory: Completeness property, Archimedean property, Denseness of rationals and irrationals, Countable and uncountable sets, Cardinality.

Metric Spaces: Open and closed sets, Interior, Closure and limit points of a set, Subspaces, Continuous functions on metric spaces, Convergence in a metric space, Complete metric spaces, Compact metric spaces, Compactness and uniform continuity, Connected metric spaces, Total boundedness, Finite intersection property.

Sequence and Series of Functions: Pointwise and uniform convergence, Cauchy criterion for uniform convergence, Weierstrass M-test, Abel's and Dirichlet's tests for uniform convergence, Uniform convergence and continuity, Uniform convergence and differentiation, Weierstrass approximation theorem.

Riemann-Stieltjes Integral: Definition and existence of Riemann-Stieltjes integral, Properties, Integration and differentiation, Fundamental theorem of calculus.

Course Learning Outcomes: The student will be able to

1. analyze different properties of $\mathbb{R}$.
2. apply properties viz. convergence, completeness, compactness etc. from the real line to metric spaces.
3. identify the difference between pointwise and uniform convergence and analyze the effect of uniform convergence on the functions with respect to continuity, differentiability and integrability.
4. determine the Riemann-Stieltjes integrability of a bounded functions.

Text/References:

- W. Rudin, Principles of Mathematical Analysis, McGraw-Hill, 3rd edition, 2013.
- G. F. Simmons, Introduction to Topology and Modern Analysis, Tata McGraw Hill, 2008.
- S. C. Malik and S. Arora, Mathematical Analysis, Wiley Eastern, 2010.
- P. K. Jain and K. Ahmad, Metric Spaces, Alpha Science Publishers, 2004.
- R. G. Bartle and D. R. Sherbert, Introduction to Real Analysis, John Wiley & Sons, 4th ed., 2010.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes etc.)	25%

PMA108 Algebra I

Credits: 3.5 [3-1-0]

Course Objectives: The course intends to impart knowledge in the areas of Group Theory and Linear Algebra with an aim of enabling students to apply the concepts learned to other areas.

Contents:

Linear Transformations: Review of vector spaces, Linear transformations, Algebra of linear transformations, Matrix representation of linear transformations, Different types of matrices, Change of basis, Rank, trace and determinant of a matrix, Dual and double dual, Transpose of a linear transformation, Linear transformations and their characteristic roots and vectors, Algebraic and geometric multiplicity, Cayley-Hamilton theorem, Minimal polynomial, Canonical forms, Diagonal form, Triangular form, Rational and Jordan form.

Group Theory: Center, Normalizer, Centralizer, Homomorphism and isomorphism, Cyclic groups and their Properties, Classification of subgroups of cyclic groups, Permutation groups, Cyclic notation of permutation groups, Cayley's theorem, Conjugate elements, Class equation, Direct product of groups, Fundamental theorem of finite abelian groups, Cauchy's theorem, Sylow theorems and its applications.

Course Learning Outcomes: The students will be able to

1. find characteristic roots, vectors, and quadratic forms of matrices.
2. recognize various canonical forms of linear transformations.
3. describe the properties of permutation groups and cyclic groups.
4. apply structure theory of abelian groups and Sylow theorems to solve different problems.

Text/References:

- K. Hoffmann and R. Kunze, Linear Algebra, Pearson, Second edition, 2014.
- I. N. Herstein, Topics in Algebra, Wiley Eastern Ltd., 2005.
- S. Singh and Q. Zameeruddin, Modern Algebra, Vikas Publishing House, 2006.
- S. Luthar and I. B. S. Passi, Algebra (Vol. 1 and 2), Narosa Publishing House, 1999.
- J. A. Gallian, Contemporary Abstract Algebra, Cengage Learning, 8th ed., 2013.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes etc.)	25%

PMA109 Ordinary Differential Equations

Credits: 3.5 [3-1-0]

Course Objectives: The main objective is to provide mathematics majors with an introduction to the theory of Ordinary Differential Equations (ODEs) through applications and methods of solution. Students will become knowledgeable about system of ODEs and how they can serve as models for physical processes. The course will also develop an understanding of the elements of analysis of ODEs.

Contents:

Qualitative Properties of Solutions: Existence and uniqueness of initial value problems for first order equations, Picard's existence theorem, non-local existence of solutions, Gronwall's inequality and uniqueness, continuation of solutions and continuous dependence, Peano's existence theorem.

Sturm Theory: Wronskian and fundamental solution, Sturm separation theorem and oscillations, Sturm comparison theorem and applications.

Boundary Value Problems: Two-point boundary value problems, Sturm-Liouville theory, Lagrange's identity, eigenvalues and eigenfunctions of the regular and periodic Sturm-Liouville problems, eigenfunction expansion, non-homogeneous problems, singular Sturm-Liouville problems, Green's function.

Linear Systems: Origin of system, linear system with constant coefficients, existence of solutions, Wronskian, fundamental solutions, Abel's formula, fundamental matrices, matrix exponential form of solution, variation of parameters, linear system with variable coefficients.

Nonlinear Systems and Stability: Phase plan, autonomous system, critical points and stability, stability by Liapunov's direct method, simple critical points, periodic solutions: Poincare-Bendixson theorem, limit cycles.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. give examples of differential equations for which either existence or uniqueness of solution fails.
2. state correctly and apply basic facts of Sturm separation and comparison theorems.
3. solve boundary value problems and Sturm-Liouville problems.
4. state correctly and apply basic facts of systems: eigen-values and eigen-vector, fundamental matrices, and non-homogeneous systems.
5. sketch the phase portraits and apply standard methods to check the stability of critical points for autonomous system.

Text/References:

- William E. Boyce, Richard C. DiPrima, and Douglas B. Meade, Elementary Differential Equations and Boundary Value Problems, 11th edition, Wiley, 2017.
- E. A. Coddington and N. Levinson, Theory of Ordinary Differential Equations, McGraw Hill Education, 2017.
- George F. Simmons and Steven G. Krantz, Differential Equations: Theory, Technique, and Practice, McGraw Hill Education, 2006.
- L. Perko, Differential Equations and Dynamical Systems, 3rd Edition, Springer, 2008.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMA110 Mechanics

Credits: 3.5 [3-1-0]

Course Objectives: This course is intended to provide a treatment of basic knowledge in mechanics used in deriving a range of important results and problems related to rigid bodies The objective is to provide the student the classical mechanics approach to solve a mechanical problem.

Contents:

Dynamics of a Particle: Tangential and normal accelerations, Simple harmonic motion, Oscillatory motion projectile motion, Central forces, Apses and apsidal distances, Stability of orbits, Kepler's laws of planetary motion, Simple pendulum, Motion in a resisting medium, Motion of a pendulum in a resisting medium.

Linear and Angular Momentum: Rate of change of angular momentum for a system of particles, Impulsive forces, Moments and products of inertia of a rigid body, Equimomental system, Principal axes, Coplanar distribution, General equations of motion.

Motion About a Fixed Axis: Compound pendulum, Motion in two dimensions, Euler's dynamical equations and simple stability considerations.

Lagrangian and Hamiltonian Mechanics: Constrained motion, D'Alemberts principle, Variational Principle, Lagrange's equations of motion, Generalized coordinates, cyclic coordinates, Hamiltons principles, Principles of least action, Hamilton's equation of motion, Phase Space, State space examples, Cannonical transformations, Lagrange's and Poisson brackets invariance.

Course Learning Outcomes: The student will be able to

1. describe the dynamics involving a single particle like projectile motion, Simple harmonic motion, pendulum motion and related problems.
2. analyze the path described by the particle moving under the influence of central force.
3. apply the concept of system of particles in finding moment inertia, directions of principle axes and consequently Eulers dynamical equations for studying rigid body motions.
4. obtain the equation of motion for mechanical systems using the Lagrangian and Hamiltonian formulations of classical mechanics.
5. obtain canonical equations using different combinations of generating functions.

Text/References:

- F. Chorlton, Text book of Dynamics, CBS Publishers, 1985.
- J. L. Synge and B. A. Griffith, Principles of Mechanics, Tata McGraw Hill, 1971.
- C. Fox, An Introduction to the Calculus of Variations, Dover Publications, 1992.
- H. Goldstein, C. Poole, and J. Safko, Classical Mechanics, Addison Wesley, 2002.
- P. Mann, Lagrangian and Hamiltonian Dynamics, Oxford University Press, 2018.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes etc.)	25%

PMA111 Computer Programming

Credits: 4 [3-0-2]

Course Objectives: This course is designed to explore computing and to show students the art of computer programming. Students will learn some of the design principles for writing good programs.

Course Outlines:
Computer Fundamentals: Introduction to computer systems, number system, integer, signed integer, fixed and floating point representations, IEEE standards, integer and floating point arithmetic; CPU organization, ALU, registers, memory, the idea of program execution at micro level.

Algorithms and Programming Languages: Algorithm, flowcharts, pseudocode, generation of programming languages.

C **Language:** Structure of *C* Program, life cycle of program from source code to executable, compiling and executing C code, keywords, identifiers, primitive data types in *C*, variables, constants, input/output statements in C, operators, type conversion and type casting, conditional branching statements, iterative statements, nested loops, break and continue statements.

Functions: Declaration, definition, call and return, call by value, call by reference, showcase stack usage with help of debugger, scope of variables, storage classes, recursive functions, recursion vs Iteration.

Arrays, Strings and Pointers: One-dimensional, two-dimensional and multi-dimensional arrays, operations on array: traversal, insertion, deletion, merging and searching, Inter-function communication via arrays: passing a row, passing the entire array, matrices. Reading, writing and manipulating Strings, Understanding computer memory, accessing via pointers, pointers to arrays, dynamic allocation, drawback of pointers.

Object Oriented Programming Concepts: Data hiding, abstract data types, classes, access control; class implementation, constructors, destructor operator overloading, friend functions; object oriented design (an alternative to functional decomposition) inheritance and composition; dynamic binding and virtual functions; polymorphism; dynamic data in classes.
Laboratory work: To implement Programs for various kinds of programming constructs in *C* Language.

Course Learning Outcomes: On successful completion of this module, students will be able to:

1. comprehend concepts related to computer hardware and software, draw flowcharts and write algorithm/pseudocode.
2. write, compile and debugprograms in *C* language, use different data types, operators and console I/O function in a computer program.
3. design programs involving decision control statements, loop control statements, case control structures, arrays, strings, pointers, functions and implement the dynamics of memory by the use of pointers.
4. comprehend the key concepts of object-oriented design and programming concepts.

Text/References:

- Y. Kanetkar, Let Us C, BPB Publications, 2nd ed., 2016.
- Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, Pearson, 2nd ed, 2015.
- H. M. Deitel and P. J. Deitel, C++ How to Program, Prentice Hall, 8th Ed, 2011.
- E. Balaguruswamy, Object Oriented Programming with C++, McGraw Hill, 2013.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC105 Discrete Mathematical Structures

Credits: 3.5 [3-1-0]

Course Objectives: This course develop mathematical foundations for explaining and creating mathematical arguments, solving problems by constructing recurrence relations and generating functions, and exploring counting problems that are often required in learning other mathematical and computer sciences courses.

Course Outlines:

Mathematical Logic: Statement and notations, Connectives, Statement formulas and truth table, Conditional and bi-conditional statements, Tautology and contradiction, Equivalence of formulas, Tautological implications.

Theory of inference: Validity using truth table, Rules of inference, Consistency of premises and indirect method of proof, Predicates, Statement function, Variables, Quantifiers, Free and bound variables, Universe of discourse, Inference of the predicate calculus.

Relation: Review of binary relations, equivalence relations, counting of reflexive and symmetric relations, Compatibility relation, Composition of binary relations, Composition of binary relations and transitive closure, Partial ordering and partial ordered set.

Function: Review of functions and their enumeration, Pigeonhole principle.

Recurrence Relation: Iteration, Recurrence relations, Generating function, solving linear recurrence relation, solving recurrence relation using generating function.

Lattice and Boolean Algebra: Lattice and algebraic system, Basic properties of algebraic systems, Special types of lattices, Distributed, Complemented lattices, Boolean algebra, Boolean expressions, Normal form of Boolean expressions, Boolean function and its applications to LOGIC GATES.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. construct mathematical arguments using logical connectives and quantifiers.
2. validate the correctness of an argument using statement and predicate calculus.
3. apply lattices and Boolean algebra as tools in the study of network.
4. explain some of the discrete structures which include sets, relations, functions, and graphs.
5. construct and solve recurrence relations with or without generating functions.

Text/References:

- J. P. Tremblay and R. Manohar, A First Course in Discrete Structures with Applications to Computer Science, McGraw Hill, 1987.
- Kenneth H. Rosen, Discrete Mathematics and its Applications, McGraw Hill Education; 7th edition, 2017.
- C. L. Liu, Elements of Discrete Mathematics, McGraw Hill, New York, 1978.
- R. P. Grimaldi and B. V. Ramana, Discrete and Combinatorial Mathematics – An Applied Introduction, Pearson Education, 5th edition, 2006.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMA204 Measure Theory and Integration

Credits: 3.5 [3-1-0]

Course Objectives: The aim of the course is to introduce the students Lebesgue measure, measurable sets and their properties, measurable functions and their properties and Lebesgue integral.

Course Outlines:

Lebesgue Measure: Introduction, Outer measure, Lebesgue measure, measurable sets, Properties of measurable sets, Borel sets and their measurability, non-measurable sets.

Measurable Functions: Definition and properties of measurable functions, step functions, characteristic functions, simple functions, Littlewood's three principles, convergence in measure.

Lebesgue Integral: Lebesgue integral of bounded function, Integration of non-negative functions, General Lebesgue integrals, Integration of series, Comparison of Riemann and Lebesgue integrals.

Differentiation and Integration: Differentiation of monotone functions, functions of bounded variation, Lebesgue differentiation theorem, differentiation of an integral, absolute Continuity.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. define Lebesgue measure on $\mathbb{R}$.
2. describe measurable functions and its properties.
3. apply measures to construct integrals and explain convergence theorems for the Lebesgue integral.
4. analyze the relation between differentiation and Lebesgue integration.

Text/References:

- H. L. Royden and P. M. Fitzpatrick, Real Analysis, Pearson Education, 4th Edition, 2011.
- G. de Barra, Measure Theory and Integration, Wiley Eastern Ltd., 2012.
- P. K. Jain and V. P. Gupta, Lebesgue Measure and Integration, New Age International Ltd., 2nd Edition, 2010.
- I. K. Rana, An Introduction to Measure and Integration, Narosa Publication House, 2nd Edition, 2010.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMA205 Algebra II

Credits: 3.5 [3-1-0]

Course Objectives: The course intends to impart knowledge in the areas of ring theory and field theory with an aim of enabling students to apply the concepts learned to other areas.

Course Outlines:

Ring Theory: Special kinds of rings, Subrings and ideals, Homomorphism and isomorphism, Quotient rings, Prime and maximal ideals, Characteristic of a ring, Integral domain, Units and zero divisors, Polynomial rings, Irreducible and prime elements, Unique factorization domain, Principal ideal domain, Euclidean domain.

Field Theory: Polynomials and their irreducibility criteria, Adjunction of roots, Finite and infinite extensions, Algebraic and transcendental extensions, Algebraically closed fields, Splitting fields, Normal extension, Multiple roots, Finite fields, Separable and inseparable extensions, Automorphism groups and fixed fields, Fundamental theorem of Galois theory.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. describe the properties of integral domain, principle ideal domain, Euclidean domain and unique factorization domain.
2. apply different irreducibility criteria to check the irreducibility of a polynomial.
3. describe the concepts of fields, various extensions of fields and splitting fields.
4. analyze the properties of finite fields and Galois theory.

Text/References:

- I. N. Herstein, Topics in Algebra, Wiley Eastern Ltd., 2005
- S. Singh and Q. Zameeruddin, Modern Algebra, Vikas Publishing House, 2006.
- J. A. Gallian, Contemporary Abstract Algebra, Cengage Learning, 8th ed, 2013.
- P. B. Bhattacharya, S. K. Jain, and S. R. Nagpaul, Basic Abstract Algebra, Cambridge University Press, 1997.
- I. S. Luthar and I. B. S. Passi, Algebra (Vol. 4), Narosa Publishing House, 2004.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMC205 Data Base Management System

Credits: 4 [3-2-0]

Course Objectives: The major objective of this course is to provide a strong formal foundation in database concepts, technology and to give an introduction to systematic database design approaches covering conceptual design, logical design and an overview of physical design. This course will also introduce the concepts of transactions and transaction processing to present the issues and techniques relating to concurrency and recovery. The overriding concern, is present the concepts and techniques by SQL engine and PL/SQL programming

Contents:

Introduction: Basic concepts, Database and database users, Characteristics of the database, Database systems concepts and architecture, Data models, Schemas and instances, DBMS architecture and data independence, Database languages and interfaces.

Data Modeling: ER model concepts, Notation for ER diagram, Mapping constraints, Concepts of primary key, Candidate key, Foreign key and super key. Relational Model, Languages and Systems: Relational data model and relational algebra, Relational model concepts, Relational model constraints, Relational calculus: Tuple and domain calculus.

SQL: SQL data types. Data definition in SQL, View, Queries and sub queries in SQL, Specifying constraints and indexes in SQL, Cursors, Triggers, Procedures and packages in PL/SQL.

Relational Data Base Design: Function Dependencies and normalization for relational databases, Normal forms: INF, 2NF, 3NF, and BCNF, Loss less join and dependency preserving decomposition.

Concurrency Control and Recovery Techniques: Concurrency control techniques, Locking techniques, Time stamping protocols for concurrency control, Multiple granularity of data items, Recovery techniques, Database backup and recovery.

Laboratory Work: Laboratory exercises should include defining schema for applications, creation of a database, writing SQL and PL/SQL queries for database operations.

Minor Projects: The minor projects will be set in consonance with material covered in theory and laboratory classes.

Learning Outcomes: The student will be able to

1. explain effectively the underlying concepts of database technologies.
2. design and implement a database schema for a given problem-domain.
3. populate and query a database using SQL DML/DDL commands.
4. programming PL/SQL including stored procedures, stored functions, cursors, packages.

Text/References:

- C. J. Date, An Introduction to Database Systems, Narosa Publishing House, 2002.
- A. Silberschatz, Henry F. Korth, and S. Sudarshan, Database System Concepts, McGraw Hill Education. 6th edition, 2013.

- B. Desai, An Introduction to Database Concepts, Galgotia Publication, 2002.
- R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, Pearson Education, 7th edition, 2017.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMA207 Complex Analysis

Credits: 3.5 [3-1-0]

Course Objectives: The course aims to introduce the theory of complex analysis to graduate students with applications to solve problems in the mathematical sciences and engineering.

Course Outlines:

Complex numbers: Introduction to complex numbers, geometrical interpretation, different representations of complex numbers, Stereographic projection.

Elementary and analytic functions: functions of complex variables, examples of elementary functions like exponential, trigonometric and hyperbolic functions, elementary calculus on the complex plane (limits, continuity, differentiability), Cauchy-Riemann equations, analytic functions, harmonic functions with examples, branch points and branch cuts, multi-valued functions (eg. logarithmic function and its branches, Riemann surfaces).

Complex integration: Cauchy's integral theorem, Cauchy integral formula for higher derivatives, Morera's theorem, Liouville's theorem, maximum-modulus principle, Schwarz lemma.

Series expansion of complex functions: Power series, Taylor and Laurent series of complex functions, convergence, definition of holomorphic and meromorphic functions, zeros and poles, classification of singular points, removable singularities, Weierstrass theorems (M test and factor theorem), argument principle and Roche's theorem (eg. with application to prove the fundamental theorem of algebra).

Residue calculus: General form of Cauchy's theorem, Cauchy residue theorem, evaluation of definite integrals using residue theorem (principal value integrals and integrals with branch points), residue at infinity.

Conformal mappings: Elementary conformal mappings (Schwarz-Christoffel transformation), analytic continuation, method of analytic continuation by power series (e.g. application in defining the Riemann-Zeta function).

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. represent complex numbers in Cartesian, polar and matrix form, geometrical interpretation of complex numbers.
2. inspect the analyticity of complex functions including the utility of Cauchy-Riemann equations, evaluation of contour integrals using Cauchy integral formula.
3. represent complex functions as power series (e.g., Taylor and Laurent) and their convergence, classification of singularities.
4. apply residue calculus using Cauchys residue theorem and method of analytic continuation.
5. have knowledge of conformal maps.

Text/References:

- M. Ablowitz and A. S. Fokas, Complex Variables: Introduction and Applications, Cambridge University Press, 2nd edition, 2003.
- R. V. Churchill and J. W. Brown, Complex Variable and Applications, McGraw Hill, 8th edition, 2009.
- L. V. Ahlfors, Complex Analysis, Tata McGraw Hill, 3rd edition, 1979.

- H. S. Kasana, Complex Variables: Theory and Applications, Prentice Hall India, 2nd edition, 2005.
- S. Ponnuswamy, Foundation of Complex Analysis, Narosa Publishing House, 2nd edition, 2011.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMA208 Numerical Analysis

Credits: 4.5 [3-1-2]

Course Objectives: The primary goal is to provide mathematics majors with a basic knowledge of numerical methods including: root finding, numerical linear algebra, interpolation, integration, solving systems of linear equations, curve fitting, and numerical solution to ordinary differential equations. MATLAB is the software environment used for implementation and application of these numerical methods. The course will also develop an understanding of the elements of error analysis for numerical methods and certain proofs. The course will further develop problem solving skills.

Course Outlines:

Mathematical Preliminaries and Error Analysis: Round off errors, Algorithms and convergence, conditioning and stability.

Numerical Solution of Nonlinear Equations: Review of bisection, fixed point and secant method, Newton's method and its variants, convergence analysis, zeros of polynomials and Muller's method, Newton's method in higher dimensions.

Solutions of Linear Systems: Direct methods, Gauss-elimination method, pivoting, matrix factorization. Iterative methods: Matrix norms, Jacobi and Gauss-Seidel, Relaxation methods and their convergence, error bounds and iterative refinement. Computation of eigenvalues and eigenvectors: power method, Householder's method, QR algorithm.

Interpolation: Lagrange interpolation, Newton's divided differences, Hermite interpolation, Splines, Richardson's extrapolation.

Numerical Integration: Newton-Cotes formula, Trapezoidal and Simpson's rules, Gaussian quadrature, Romberg integration, Multiple integrals.

Numerical Solution of Ordinary Differential Equations: Initial value problems, Euler's method, Higher-order methods of Runge-Kutta type, Multi-step methods, Adams-Bashforth, Adams-Moulton and Milne's methods, convergence and stability analysis, system of equations. Boundary value problems: Shooting methods, finite differences, Rayleigh-Ritz methods.

Lab Experiment: Implementation of numerical techniques using MATLAB based on course contents.

Course Learning Outcomes: On successful completion of this module, students will be able to:

1. find the source of errors and its effect on any numerical computations and be familiar with finite precision computations.
2. solve an algebraic or transcendental equation using an appropriate numerical method and perform an error analysis for a given numerical method.
3. solve a linear system of equations using an appropriate numerical method which include direct and iterative methods and apply numerical methods to find eigen-values and corresponding eigen-vectors.
4. approximate the given data with an appropriate interpolating polynomial.
5. calculate a definite integral numerically and solve initial and boundary value problems using appropriate numerical methods.

Text/References:

- Richard L. Burden, J. Douglas Faires and Annette Burden, Numerical Analysis, Cengage Learning, 10th edition, 2015.

- E. Ward Cheney and David R. Kincaid, Numerical Mathematics and Computing, Cengage Learning, 7th edition, 2012.
- K. Atkinson and W. Han, Elementary Numerical Analysis, John Willey and Sons, 3rd edition, 2004.
- Endre Suli and David F. Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC212 Data Structures and Algorithms

Credits: 5 [3-0-4]

Course Objectives: This course shall help the students in explaining how to analyze a given algorithm. This course will also help them in making the choice of data structures and methods to design algorithms that affect the performance of programs. They will also learn to solve problems using data structures such as linear lists, stacks, queues, etc. and shall be writing programs for these structures

Contents:

Introduction: Need and definition of data structures, Asymptotic notations, Recursion and recursive functions.

Fundamental Data Structures: Arrays, Stacks, Queues, Linked lists and trees.

Searching and Sorting: Linear search, Binary search, Insertion-sort, Bubble-sort, Selection-sort, Merge sort, Heap sort, Priority queue, Quick sort, Sorting in linear time, Hash tables, Binary search tree.

Graphs: Elementary graph algorithms, Minimum spanning tree, Shortest path algorithms.

Algorithm Design Techniques: Divide and conquer, Greedy algorithm.

Laboratory Work: Laboratory exercises will be set in consonance with the material covered in lectures. This will include assignments in a programming language like C and C++ in GNU Linux environment.

Minor Projects: The minor projects will be set in consonance with material covered in theory and laboratory classes

Learning Outcomes: The student will be able to

1. analyze a given algorithm for its complexity.
2. reflect the role of data structures in different implementations.
3. implement algorithm design techniques to different problems.

Text/References:

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, Prentice-Hall of India, 2007.
- M. T. Goodrich and R. Tamassia, Data Structures and Algorithms in Java, Wiley, 2006.
- A. V. Aho, J. E. Hopcroft, and J. D. Ullman, Data Structures and Algorithms, Pearson, 1982.
- S. Sahni, Data Structures, Algorithms and Applications in C++, Silicon Press, 2005.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

Third Semester

PMA301 Functional Analysis

Credits: 3.5 [3-1-0]

Course Objectives: Functional analysis is a fundamental area of pure mathematics, with countless applications to the theory of differential equations, engineering, and physics. The students will be exposed to the theory of Banach space, Hilbert spaces, linear transformations and functionals. In particular, the major theorems in functional analysis, namely, Hahn-Banach theorem, uniform boundedness theorem, open mapping theorem and closed graph theorem will be covered.

Course Outlines:

Review of some basic concepts in metric spaces and topological spaces, completeness proofs and completion of metric spaces.

Normed linear spaces and Banach spaces, examples of Banach spaces, quotient spaces, equivalent norms, finite dimensional Banach spaces and compactness, bounded and continuous linear operators and linear functionals, dual space, Banach fixed-point theorem and applications, Hahn-Banach theorem and applications, uniform boundedness theorem, open mapping and closed graph theorem.

Inner product spaces and its properties, Hilbert spaces and examples, best approximation in Hilbert spaces, orthogonal complements, orthonormal basis, Gram Schmidt orthogonalisation, dual of a Hilbert space.

Operator theory, adjoint of an operator, Riesz representation theorem, self-adjoint operators, normal and unitary operators, projection operator.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. recognize the fundamentals of normed linear spaces and completeness property.
2. understand the fundamentals of Hilbert spaces and can provide best approximations in Hilbert spaces.
3. represent the functionals/dual of a normed linear space and inner product space.
4. define and thoroughly explain self-adjoint, normal and unitary operators and analyze operators from applications.

Text/References:

- Erwin Kreyszig, Introductory Functional Analysis with Applications, Wiley, 2007.
- John B. Conway, A course in Functional Analysis, Springer, 2nd edition, 1990.
- G. F. Simmons, Introduction to Topology and Modern Analysis, McGraw Hill Education, 2017.
- S. Kesavan, Functional Analysis, Hindustan Book Agency, 2014.
- Peter D. Lax, Functional Analysis, John Wiley & Sons, 2002.
- W. Rudin, Functional Analysis, McGraw Hill, 1991.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

PMC206 Operating Systems

Credits: 4 [3-0-2]

Course Objectives: The course aims to shape the attitudes of learners regarding the field of computer organization as well as operating system. Specifically, the course aim to i) have the insight of computer organizations and the working of operating systems ii) instill the belief that computer organization as well as operating is important for IT professional.

Contents:

Introduction: Computer organization, Operating system, Types of operating systems, Register transfer language, Overview of data representation in computer, Instruction, Instruction codes, Instruction types, Instruction set completeness, Instruction cycle, Execution cycle, Addressing modes, Control unit: Micro programmed control Vs hardwired control, RISC Vs CISC.

Input Output Organization: Input and output interface, Asynchronous data transfer, Modes of transfer, DMA, I/O interrupts, Channels.

Process management: Processor scheduling, Schedulers, CPU scheduling algorithms, Concurrent process: Introduction to conflicts due to concurrency, Critical section problem, Deadlock: Introduction to deadlock prevention and avoidance, Detection and recovery.

Memory Management: Memory hierarchy, Associative memory, Demand paging-virtual memory and Segmentation, File Management: File system structure, Allocation methods, Secondary storage management-disk scheduling.

Laboratory Work: Implementation of CPU scheduling algorithms, process management, memory management and file management functions of Linux/Unix operating systems.

Learning Outcomes: The student will be able to

1. explain effectively the underlying concepts of computer organization.
2. describe instruction set, instruction cycle, RISC as well as CISC.
3. apply the process management, scheduling and deadlock.
4. analyse the memory management, and file system.

Text/References:

- M. Morris, Computer System Architecture, Pearson Education, 2007.
- A. Silberschatz, Peter B. Galvin, and G. Gagne, Operating Systems Concepts, Willey, 8th edition, 2017.
- John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach,Morgan Kaufmann, 6th edition, 2017.
- W. Stallings, Computer Organization and Architecture: Designing for Performance, Pearson Education, 9th edition, 2013.
- D. Dhamdhere, Operating Systems: A Concept-Based Approach, McGraw Hill Education, 3rd edition, 2017.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	40%
Sessionals (Assignments/Quizzes/Lab Evaluation)	35%

PMA303 Probability and Statistics

Credits: 4.5 [3-1-2]

Course Objectives: The course aims to shape the attitudes of learners regarding the field of statistics. Specifically, the course aims to i) motivate students towards an intrinsic interest in statistical thinking, ii) instill the belief that statistics is important for scientific research.

Course Outlines:

Introduction & Random variables: Classical and Empirical Probability, Axioms of probability, probability space, conditional probability, independence, Baye's rule, Random variable, some common discrete and continuous distributions (Binomial, Poisson, Geometric, Negative binomial, Rectangular, Exponential, Normal, Gamma).

Mathematical Expectation: Moments, Moment generating functions, Characteristic function and properties.

Bivariate Probability Distribution: Probability distribution of functions of a random variable, joint and marginal distributions, conditional distributions.

Limit Theorems: Modes of convergence, Markov and Chebyshev's inequalities, Law of large numbers, Central limit theorem.

Correlation and Regression: Covariance, Karl-Pearson and rank Correlation coefficients; linear regression between two variables.

Hypothesis tests: Introduction to Sampling Distribution (standard normal, chi-square, t & F distributions), Theory of Estimation, Properties of an estimator, Tests for Goodness of fit, Method of maximum likelihood, Method of moments, Critical regions, Type I and II errors, Neyman-Pearson lemma (without proof).

Parametric & Non-parametric tests: T-test, Z-test, Chi-square test, Sign Test, Wilcoxon Signed-rank Test, Kolmogorov Smirnov Test.

Laboratory Work: Lab work will be based on the programming in MATLAB/SPSS/R language of various statistical techniques.

Minor Projects: The minor projects will be set in consonance with material covered in theory and laboratory classes.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. compute probabilities of events along with an understanding of random variables and distribution functions and expectation.
2. understand the convergence of sequence in probabilities
3. analyze the correlated data and fit the linear regression models.
4. make statistical inferences using principles of hypothesis tests.

Text/References:

- P. L. Meyer, Introduction to Probability and Statistical Applications, Oxford & IBH, 2007.
- A. M. Goon, M. K. Gupta, and B. Dasgupta, An Outline of Statistical Theory, Vol. I, The World Press Pvt. Ltd., 2000.

- R. V. Hogg, and A. T. Craig, Introduction to Mathematical Statistics, Prentice Hall of India, 2004.
- R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, Probability and Statistics for Engineers and Scientists, Pearson, 2010.
- R. A. Jhonson, C. B. Gupta, Miller and Freund's Probability and Statistics for Engineers, Dorling Kindersley, 7 ed., 2007

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMA304 Mathematical Programming

Credits: 4.5 [3-1-2]

Course Objectives: Operations research helps in solving problems in different environments that needs decisions. This module aims to introduce students to use quantitative methods and techniques for effective decisionsmaking; model formulation and applications that are used in solving business decision problems.

Course Outlines:

Linear programming: Linear programs formulation through examples from engineering / business decision making problems, preliminary theory and geometry of linear programs, basic feasible solution, simplex method, variants of simplex method, like two phase method and revised simplex method; duality and its principles, interpretation of dual variables, dual simplex method, primal-dual method.

Integer programming problems: Linear integer programs, their applications in real decision making problems, cutting plane and branch and bound methods.

Transportation and Assignment problem: Initial basic feasible solutions of balanced and unbalanced assignment/transportation problems, optimal solutions, time minimization assignment/transportation problem. Game Theory: Two person zero-sum game, Game with mixed strategies, Dominance property, solution by linear programming.

Nonlinear Programming: Concepts of convexity and its generalizations, Maxima and minima of convex functions, unconstrained optimization problems, constrained programming problems, Lagrange's multiplier rule and Kuhn-Tucker conditions for constrained programming problems. quadratic programs, Wolfe method, applications of quadratic programs in some domains like portfolio optimization and support vector machines, etc.

Laboratory Work: Laboratory experiments swill be set in consonance with the materials covered in theory.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. formulate and solve some real life problems as Linear programming problem.
2. use the simplex method to find an optimal vector for the standard linear programming problem and the corresponding dual problem.
3. find optimal solution of transportation problem and assignment problem
4. solve two person zero sum game.
5. solve unconstrained and constrained nonlinear programming problems.

Text/References:

- S. Chandra, Jayadeva, and A. Mehra, Numerical Optimization and Applications, Narosa Publishing House, 2013.
- R. W. Cottle and M. N. Thapa, Linear and Nonlinear Optimization, Springer, 2017.
- M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, Linear Programming and Network Flows, John Wiley and Sons, 1990.
- M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley and Sons, 1993.

- M. Ferris, O. Mangasarian, and S. Wright, Linear Programming with Matlab, MPS-SIAM series on Optimization, 2007.
- G. Hadley, Linear Programming, Narosa Publishing House, 2002.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC303 Computer Networks

Credits: 4[3-0-2]

Course Objectives: The course has been intended to impart knowledge in the domain of topology, data communication, protocols and data propagation issues in computer networks. The contents are descriptive to enforce knowledge of working of seven layers of network model, path finding issues, security and other communication paradigm.

Contents:

Introduction to Computer Networks: Advantages of networks, Type of networks, Network topologies, Protocol hierarchy, OSI reference model, TCP/IP reference model, Comparison of OSI and TCP/IP reference model. Standard organizations, Data rate limits, Digital and analog transmission, Transmission media, Multiplexing and switching Techniques, Network devices.

Data Link Layer: Design issues, Framing, Error control, Flow control, Error correcting codes, Error detecting codes, Sliding Windows protocols, HDLC, Channel allocation problem, Multiple access protocols, Ethernet.

Network Layer: Network layer design issues, Routing algorithms: Shortest path algorithm, flooding, Distance vector routing, Link state routing, Hierarchical routing, Broadcast routing, multicast routing, Congestion control algorithms, Internetworking, IP protocol, IP addressing, Subneting, Superneting, ARP, RARP, BOOTP, DHCP, ICMP, OSPF, BGP, Internet multicasting, Mobile IP, Ipv6.

Transport Layer: Services provided to the upper layer, Addressing, UDP, TCP, Connection establishment, connection release, Flow control and buffering, Multiplexing, TCP congestion control, TCP timer management.

Application Layer: Introduction to Domain name system, E-mail, File transfer protocol, HTTP, HTTPS, World Wide Web.

Laboratory Work: The lab work will be based on network installation using Windows server and Linux server, implementation of the different application layer protocols on these server, configuration of network devices like switches, routers and implementation of the protocols on these devices. Socket programming using different types of sockets.

Learning Outcomes: The student will be able to

1. describe the functions and working of different networking devices.
2. explain topology, data communications, protocol and data propagation issues in computer networks.
3. know the working of seven layer network model, TCP/IP, OSI etc.
4. describe the different routing algorithms.

Text/References:

- B. A. Forouzan, Data Communications and Networking, Tata McGraw Hill, 2006.
- A. S. Tannenbaum, Computer Networks, Pearson Education, 2004.
- W. Stalling, Data and Computer Communication, Pearson Education, 2003.

- J. L. Antonako, K. C. Mansfield, An Introduction to Computer Networks, Pearson Education, 2002.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	40%
Sessionals (Assignments/Quizzes/Lab Evaluation)	35%

Fourth Semester

PMA401 Number Theory

Credits: 3.5 [3-1-0]

Course Objectives: The purpose of this course is to introduce classical number theory, primes, congruences, solution of congruences, arithmetic functions, quadratic residues and primitive roots. Apart from teaching the theory, stress will be on solving problems and helping students to understand applications of number theory in cryptography and in other fields.

Course Outlines:

Divisibility and Congruences: Least and greatest common divisor, Fundamental theorem of arithmetic, congruence, residue classes, Chinese remainder theorem, congruences with prime moduli, Fermat's theorem, Euler's theorem and Wilson Theorem, applications to cryptography.

Arithmetic Functions: $\phi(x)$, $d(x)$, $\mu(x)$, $\sigma(x)$, Mobius inversion formula and greatest integer function, the symbol small "oh", big"oh" and their basic properties. Perfect numbers, Mersenne primes and Fermat numbers.

Polynomial Congruences: Primitive roots, indices and their applications, Quadratic residues, Legendre symbol, Euler's criterion, Gauss's Lemma, Quadratic reciprocity law, Jacobi symbol, Binary quadratic forms and their reduction, sums of two and four squares, positive definite binary quadratic forms.

Diophantine Equations: Sums of two and four squares, Pell's equations and Fermat's Last Theorem (statement only).

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. apply the properties of divisibility and prime numbers to prove results.
2. solve the linear congruences and apply in cryptography.
3. prove results using the properties of arithmetic functions.
4. implement the theory to solve polynomial congruences.
5. analyze the integral solutions of system using Diophantine equations and sum of squares.

Text/References:

- I. Niven, S. H. Zuckerman, and L. H. Montgomary, An Introduction to Theory of Numbers, John Wiley and Sons, 1991.
- H. Davenport, Higher Arithmetic, Cambridge University Press, 1999.
- David M. Burton, Elementary Number Theory, Wm. C. Brown Publishers, Dubuque, Ivova, 1989.
- T. M. Apostol, Introduction to Analytic Number Theory, Springer-Verlag, 1998.
- Hardy and W. H. Wright, Theory of Numbers, Oxford University Press, 1979.
- J. B. Dence and T. P. Dence, Elements of the Theory of Numbers, Academic Press, 1999.
- Johannes A. Buchmann, Introduction to Cryptography, Springer Verlag, 2001.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes etc.)	25%

PMA402 Mathematical Methods

Credits: 3.5 [3-1-0]

Course Objectives: This course isintendedto prepare the studentwithmathematicaltools and techniques that are required in advanced coursesoffered in the applied mathematics and engineering programs. The objective of this course is to enable students to apply transforms and variation problem technique for solving differential equations and extremum problems.

Course Outlines:

Laplace Transform: Review of Laplace transform, Applications of Laplace transform in initial and boundary value problems, Heat equation, Wave equation, Laplace equation.

Fourier Series and Transforms: Definition, Properties, Solutions of differential equations using Fourier series. Fourier integral theorem, Convolution theorem and Inversion theorem, Discrete fourier transforms (DFT), Relationship of FT and fast Fourier transforms (FFT), Linearity, Symmetry, Time and frequency shifting, Convolution and correlation of DFT. Applications of FT to heat conduction, Vibrations and potential problems, Z-transform, Parseval's theorems.

Integral Equations: Linear integral equations of the first and second kind of fredholm and volterra type, Conversion of linear ordinary differential equations into integral equations, Solutions by successive substitution and successive approximation, Neumann series and resolvent kernel methods.

Calculus of Variation: The extrema of functionals, The variation of a functional and its properties, Euler equations in one and several independent variables, Field of extremals, Sufficient conditions for the extremum of a functional conditional extremum, Moving boundary value problems, Initial value problems, Ritz method.

Course Learning Outcomes:

On successful completion of this module, students will be able to:

1. solve initial and boundary value problems using Laplace transformation.
2. apply Fourier transformation and Z transformation to the relevant problems.
3. solve initial and boundary value problems using Fourier series.
4. find solutions of linear integral equations of first and second type (Volterra and Fredhlom).
5. obtain solution of initial and boundary value problems using theory of calculus of variations.

Text/References:

- G. F. Simmons, Differential Equations with Applications and Historical Notes, Tata McGraw Hill, 1991.
- I. M. Gelfand and S. V. Fomin, Calculus of Variations, Prentice Hall, 1963.
- Ram P. Kenwal, Linear Integral Equations: Theory and Techniques, Academic Press, 1971.
- I. N. Sneddon, The Use of Integral Transforms, Tata McGraw Hill, 1985.

Evaluation Scheme:

Mid-Semester Examination	30%
End-Semester Examination	50%
Sessionals (Assignments/Quizzes etc.)	20%

Elective I

PMC328 Computer & Network Security

Credits: 4[3-0-2]

Course Objectives: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools.

Contents:

Introduction: Security Attacks, Security Services, Security Mechanisms and Principles, Security goals, Malicious software, Worms, Viruses, Trojans, Spyware, Botnets.

Basic of Cryptography: Symmetric and asymmetric cryptography, cryptographic hash functions, authentication and key establishment, Message Authentication Codes (MACs), digital signatures, PKI.

Security Vulnerabilities: DoS attacks, Buffer Overflow, Race Conditions, Access Control Problems, Spoofing and Sniffing attacks, ARP Poisoning, Social Engineering and countermeasures.

Internet Security: TCP/IP Security, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS, Secure Shell (SSH), IPsec, Email Security, DNS Security, DNSSEC, Authentication Protocols.

Web Security: Phishing attack, SQL Injection, Securing databases and database access, Cross Site Scripting Attacks, Cookies, Session Hijacking, E-commerce security.

System Security: Firewalls, Types: Packet filter (stateless, stateful), Application layer proxies, Firewall Location and Configurations, Intruders, Intrusion Detection System, Anomaly and misuse detection.

Wireless Network Security: IEEE 802.11i Wireless LAN Security, Wireless Application Protocol Overview, Wireless Transport Layer Security, WAP End-to-End Security.

Laboratory Work: Insert malicious shell code into a program file and check its malicious or benign status, create Client Server program to send data across systems as two variants clear text data and encrypted data with different set of encryption algorithms, demonstrate Buffer Overflow and showcase EIP and other register status, perform ARP poisoning, SQL Injection and demonstrate its countermeasure methods, implement stateful firewall using IPTables, showcase different set of security protocol implementation of Wireless LAN.

Course Learning Outcomes: The student will be able to

1. comprehend and implement various cryptographic algorithms to protect the confidential data.
2. identify network vulnerabilities and apply various security mechanisms to protect networks from security attacks.
3. apply security tools to locate and fix security leaks in a computer network/software.
4. secure a web server and web application.
5. configure firewalls and Intrusion Detection System.

Text/References:

- W. Stallings, Network Security Essentials, Prentice Hall, 6th edition, 2017.

- R. W. Cheswick, M. S. Bellovin, and D. A. Rubin, Firewalls and Internet Security, Addison Wesley Professional, 2nd edition, 2003.
- K. Graves, Certified Ethical Hacking Study Guide, Sybex, 2010.
- W. Stallings, Cryptography and Network Security, Prentice Hall, 6th edition, 2013.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC329 Parallel and Distributed Computation

Credits: 4[3-0-2]

Course Objectives: To introduce the fundamentals of parallel and distributed programming and application development in different parallel programming environments.

Contents:

Parallelism Fundamentals: Scope and issues of parallel and distributed computing, Parallelism, Goals of parallelism, Parallelism and concurrency, Multiple simultaneous computations, Programming Constructs for creating Parallelism, communication, and coordination. Programming errors not found in sequential programming like data races, higher level races, lack of liveness.

Parallel Architecture: Architecture of Parallel Computer, Communication Costs, parallel computer structure, architectural classification schemes, Multicore processors, Memory Issues : Shared vs. distributed, Symmetric multiprocessing (SMP), SIMD, vector processing, GPU, co-processing, Flynn's Taxonomy, Instruction Level support for parallel programming, Multiprocessor caches and Cache Coherence, Non-Uniform Memory Access (NUMA).

Parallel Decomposition and Parallel Performance: Need for communication and coordination / synchronization, Scheduling and contention, Independence and partitioning, Task-Based Decomposition, Data Parallel Decomposition, Actors and Reactive Processes, Load balancing, Data Management, Impact of composing multiple concurrent components, Power usage and management. Sources of Overhead in Parallel Programs, Performance metrics for parallel algorithm implementations, Performance measurement, The Effect of Granularity on Performance Power Use and Management, Cost Performance trade-off.

Distributed Computing: Introduction: Definition, Relation to parallel systems, synchronous vs asynchronous execution, design issues and challenges, A Model of Distributed Computations, A Model of distributed executions, Models of communication networks, Global state of distributed system, Models of process communication.

Communication and Coordination: Shared Memory, Consistency, Atomicity, Message-Passing, Consensus, Conditional Actions, Critical Paths, Scalability, cache coherence in multiprocessor systems, synchronization mechanism.

CUDA programming model: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on the parallel computing device, API function to transfer data to parallel computing device, Concepts of Threads, Blocks, Grids, Developing kernel function that will be executed by threads in the parallelized part, Launching the execution of kernel function by parallel threads, transferring data back to host processor with API function call.

Parallel Algorithms design, Analysis, and Programming: Parallel Algorithms, Parallel Graph Algorithms, Parallel Matrix Computations, Critical paths, work and span and relation to Amdahl's law, Speed-up and scalability, Naturally parallel algorithms, Parallel algorithmic patterns like divide and conquer, map and reduce, Specific algorithms like parallel Merge Sort, Parallel graph algorithms, parallel shortest path, parallel spanning tree, Producer-consumer and pipelined algorithms.

Laboratory work: To implement parallel programming using CUDA with emphasis on developing applications for processors with many computation cores, mapping computations to parallel hardware, efficient data structures, paradigms for efficient parallel algorithms.

Course Learning Outcomes: The student will be able to

1. apply the fundamentals of parallel and distributed computing including parallel architectures and paradigms.
2. apply parallel algorithms and key technologies.
3. develop and execute basic parallel and distributed applications using basic programming models and tools.
4. analyze the performance issues in parallel computing and trade-offs.

Text/References:

- C. Lin and L. Snyder, Principles of Parallel Programming, Addison-Wesley, 2009.
- A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, Addison Wesley, 2nd edition, 2003.
- B. Gaster, L. Howes, D. Kaeli, P. Mistry, and D. Schaa, Heterogeneous Computing With OpenCL, Elsevier, 2nd edition, 2012.
- T. G. Mattson, B. Sanders, and B. Massingill, Patterns for Parallel Programming, Addison-Wesley, 2004.
- M. J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2003.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC330 Interactive Computer Graphics

Credits: 4[3-0-2]

Course Objectives: This course will introduce various interactive tools developed in computer graphics research field with their design rationales and algorithms.

Contents:

Fundamentals of Computer Graphics: Applications of computer Graphics in various fields, importance of computer graphics in different areas, Evolution of computer Graphics.

Graphics Systems: Video Display Devices, Random scan displays, raster scan displays, DVST, Flat Panel displays, Virtual reality systems, I/O Devices.

Graphics Primitives: Algorithms for drawing Line, circle, ellipse, arcs & sectors Boundary Fill & Flood Fill algorithm, Line attributes, Character attributes, Color Tables, Antialiasing lines & area boundaries.

Transformations: 2D & 3D Scaling, Translation, rotation, shearing & reflection, Composite transformation, Window to View port transformation.

Clipping: Cohen Sutherland, Liang Barsky, Nicholl-Lee- Nicholl Line clipping algorithms, Sutherland Hodgeman, Weiler Atherton Polygon clipping algorithm, Clipping in 3D.

Three Dimensional Object Representations: Modeling transformations, Interactive Picture construction techniques, Parallel & Perspective projection, Curved lines & Surfaces, Spline representations, Spline specifications, Bezier Curves & surfaces, B-spline curves & surfaces, Rational splines, Displaying Spline curves & surfaces.

Visible Surface Determination: Algorithms for visible line determination, z-buffer algorithm, list priority algorithm, scan-line algorithm, visible ray tracing, and painters algorithm, subdivision Algorithm.

Illumination Models & **Surface Rendering Methods:** Light sources, Illumination models, Surface rendering methods.

Basics Of Multimedia: Animation basics, multimedia applications, audio video formats.

Laboratory Work: Writing programs for implementing basic algorithms for line, circle etc, 2D and 3D transformations, clipping, 3D representations, algorithms for back face detection procedure, depth buffer method to display the visible surfaces in a scene containing any number of polyhedrons, A-buffer algorithm to display a scene containing both opaque and transparent surfaces, depth sorting method to display the visible surfaces in a scene containing several polyhedrons.

Course Learning Outcomes: The student will be able to

1. comprehend the concepts related to basics of computer graphics and visualization.
2. understand various graphics primitives and 2-D, 3-D geometric transformations.
3. grasp the concepts related three dimensional object representations.

Text/References:

- D. Hearn and P. Baker, Computer Graphics, Pearson, 2002S.

- J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer Graphics: Principles and Practice, Addison Wesley, 2nd edition, 1997.
- D. Rogers and J. A. Adams, Mathematical Elements for Computer Graphics, McGraw Hill, 2017.
- W. Newman and R. Sproull, Principles of Interactive Computer Graphics, McGraw Hill, 2001.
- Z. Xiang and R. A. Plastock, Schaum's Outlines of Computer Graphics, McGraw Hill, 2nd edition, 2000.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC331 Machine Learning and Applications

Credits: 4 [3-0-2]

Course Objectives: This course provides a broad introduction to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications.

Contents:

Introduction: Basic concepts, designing a learning model, types of machine learning: supervised, unsupervised, semi-supervised and reinforcement learning.

Data Pre-processing: Need of data pre-processing, data pre-processing methods: data cleaning, data integration, data transformation, data reduction, feature scaling-normalization and standardization.

Regression: Basic concepts, simple linear regression, multi-linear regression, polynomial regression, classification model evaluation parameters.

Classification: Need and applications of classification, logistic regression, Bayesian learning, decision tree using entropy and Gini index; K-nearest neighbors, support vector machine, classification model evaluation parameters.

Clustering: Need and applications of clustering, K-Means clustering, Hierarchical clustering.

Artificial Neural Network: Need and application of artificial neural network, forward and backward propagation, activation functions.

Genetic Algorithms: Basic concepts, Gene representation and fitness function, selection, recombination, mutation and elitism.

Laboratory Work: Implement data preprocessing, simple linear regression, multiple linear regression, decision tree, random forest classification, Naive Bayes algorithm; K-nearest neighbors, support vector machine, K-Means, ANN, and GA in Python/R/MATLAB.

Course Learning Outcomes: The student will be able to

1. pre-process data, apply regression models on the datasets.
2. apply decision tree learning, Bayesian learning and artificial neural network in real world problems.
3. apply genetic algorithms to optimize real world problems.
4. apply clustering models to the real world data.
5. compare different learning models and algorithms and utilize existing machine learning algorithms to design new algorithms.

Text/References:

- T. Mitchell, Machine Learning, McGraw Hill, 1997.
- E. Alpaydin, Introduction to Machine Learning, MIT Press, 3rd edition, 2014.

- C. M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2nd edition, 2011.
- D. Michie, J. D. Spiegelhalter, C. C. Taylor, and J. Campbell, Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

Elective II

PMC434 Data Analysis and Visualization

Credits: 4[3-0-2]

Course Objectives: To learn the analysis of various types of data and its visualization using visualization tools.

Contents:

Data Representation: Data Objects and Attribute Types: Nominal, Binary, Ordinal, Numeric, Discrete and Continuous, Types of data: Record, Temporal, Spatial Temporal, Graph, Unstructured and Semi structured data, Basic Statistical Descriptions of Data.

Introduction to Data Analysis: Probability and Random Variables, Correlation, Regression.

Data Analysis Pipeline: Data pre-processing- Attribute values, Attribute transformation, Sampling, Dimensionality reduction: PCA, Eigen faces, Multidimensional Scaling, Non-linear Methods, Graph based Semi-supervised Learning, Representation Learning Feature subset selection, Distance and Similarity calculation.

Data Mining Techniques for Analysis: Classification: Decision tree induction, Bayes classification, Rule-based classification, Support Vector Machines, Classification Using Frequent Patterns, k-Nearest Neighbor, Fuzzy-set approach Classifier, Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering.

Visualization: Traditional Visualization, Multivariate Data Visualization, Principles of Perception, Color, Design, and Evaluation, Text Data Visualization, Network Data Visualization, Temporal Data Visualization and visualization Case Studies.

Laboratory work: Implementation of various data analytics techniques such as classification clustering on real world problems using R.

Course Learning Outcomes: The student will be able to

1. analyze and extract features of complex datasets.
2. evaluate and visualize inter-dependencies among variables in dataset.
3. apply techniques for classification and clustering in datasets.
4. develop and validate models for real life datasets.

Text/References:

- J. Han, M. Kamber, and J. Pei, Data Mining Concepts and Techniques, Morgan Kaufmann, 3rd edition, 2011.
- R. D. Peng, R Programming for Data Science, Lulu.com, 2012.
- T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2nd edition, 2009.
- P. Simon, The Visual Organization: Data Visualization, Big Data, and the Quest for Better Decisions, John Wiley & Sons, 2014.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC435 Principles of Secure Coding

Credits: 4 [3-0-2]

Course Objectives: This course aims to provide an understanding of the various security attacks and knowledge to recognize and remove common coding errors that lead to vulnerabilities. It gives an outline of the techniques for developing a secure application.

Contents:

Introduction: Security, CIA Triad, Viruses, Trojans, and Worms, Security Concepts- exploit, threat, vulnerability, risk, attack. Decipher journey starting from FQDN to html page getting served to browser, Authoritative reply, revisit layer 2 and layer 3 of TCP/IP, DNS poisoning, ARP poisoning, C language obfuscation. ARP poisoning and its countermeasures. Buffer Overrun- Stack overrun, Heap Overrun, Array Indexing Errors, Format String Bugs, PE Code injection.

Malware Terminology: Rootkits, Trapdoors, Botnets, Key loggers, Honeypots. Active and Passive Security Attacks. IP Spoofing, Tear drop, DoS, DDoS, XSS, SQL injection, Smurf, Man in middle, Format String attack.

Types of Security Vulnerabilities: Buffer overflows, Invalidated input, race conditions, access-control problems, weaknesses in authentication, authorization, or cryptographic practices. Access Control Problems.

Need for secure systems: Proactive Security development process, Secure Software Development Cycle(SSDLC) , Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code - Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline.

Threat modelling process and its benefits: Identifying the Threats by Using Attack Trees and rating threats using DREAD, Risk Mitigation Techniques and Security Best Practices. Security techniques, authentication, authorization. Defence in Depth and Principle of Least Privilege.

Secure Coding Techniques: Protection against DoS attacks, Application Failure Attacks, CPU Starvation Attack.

Database and Web-specific issues: SQL Injection Techniques and Remedies, Race conditions, Time of Check Versus Time of Use and its protection mechanisms. Validating Input and Interprocess Communication, Securing Signal Handlers and File Operations. XSS scripting attack and its types-Persistent and Non-persistent attack XSS Countermeasures and Bypassing the XSS Filters.

Laboratory Work: In this Lab, the student will be able to practically understand how all the security attacks does has happened, as well as learn to recognize and remove common coding errors that lead to vulnerabilities. This lab also gives an outline of the techniques for developing a secure application code that consists of using network monitoring tools, implementing different types of attacks and some protection schemes. Evaluation will be mainly based on projects and assignments.

Course Learning Outcomes: The student will be able to

1. implement ARP poisoning attack and demonstrate countermeasure against these for different operating environments.

2. implement DNS poisoning attack and demonstrate authoritative reply in this context.

3. implement PE Code injection and demonstrate control hijacking via EIP manipulation
4. demonstrate skills needed to deal with common programming errors and develop secure applications.
5. demonstrate client-side attacks and identify nature of threats to software and incorporate secure coding practices throughout the planning and development of software product.
6. demonstrate SQL, XSS attack and suggest countermeasures for the same.

Text/References:

- M. Howard and D. LeBlanc, Writing Secure Code, Howard, Microsoft Press, 2nd edition, 2002.
- J. Deckard, Buffer Overflow Attacks: Detect, Exploit, Elsevier, 2005.
- F. Swiderski and W. Snyder, Threat Modeling, Microsoft Professional, 2004.
- J. C. Salt, SQL Injection Attacks and Defence, Elsevier, 2009.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC436 GPU Programming

Credits: 4[3-0-2]

Course Objectives: To study architecture and capabilities of modern GPUs and learn programming techniques for the GPU such as CUDA programming model.

Contents:

Introduction: Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding Up Real Applications, Parallel Programming Languages and Models.

History of GPU Computing: Evolution of Graphics Pipelines, The Era of Fixed-Function Graphics Pipelines, Evolution of Programmable Real-Time Graphics, Unified Graphics and Computing Processors, GPGPU, Scalable GPUs, Recent Developments, Future Trends.

Introduction to Data Parallelism and CUDA C: Data Parallelism, CUDA Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading.

Data-Parallel Execution Model: CUDA Thread Organization, Mapping Threads to Multidimensional Data, Matrix-Matrix Multiplication-A More Complex Kernel, Synchronization and Transparent Scalability, Assigning Resources to Blocks, Thread Scheduling and Latency Tolerance.

CUDA Memories: Importance of Memory Access Efficiency, CUDA Device Memory Types, A Tiled Matrix Multiplication Kernel, Memory as a Limiting Factor to Parallelism.

An Introduction to OpenCL: Data Parallelism Model, Device Architecture, Kernel Functions, Device Management and Kernel Launch, Electrostatic Potential Map in OpenCL.

Parallel Programming with OpenACC: OpenACC Versus CUDA C, Execution Model, Memory Model, Basic OpenACC Programs, Parallel Construct, Loop Construct, Kernels Construct, Data Management, Asynchronous Computation and Data Transfer.

Laboratory work: Practice programs using CUDA, OpenCL and OpenACC.

Course Learning Outcomes: The student will be able to

1. comprehend commonly used terms in parallel computing.
2. recognize common GPU architectures and Programming Models.
3. implement algorithms efficiently for common application kernels.
4. develop efficient parallel algorithms to solve given problems.

Text/References:

- J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General Purpose GPU Purpose GPU Programming, Addison-Wesley Professional, 4th Edition, 2012.
- D. B. Kirk, and W. M. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann, 3rd Edition, 2016.
- W. M. Hwu, A GPU Computing Gems Emerald Edition (Applications of GPU Computing Series), Morgan Kaufmann, 2011.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%

PMC437 Computer Vision: Algorithms and Applications

Credits: 4 [3-0-2]

Course Objectives: To understand the basic concepts of Computer Vision. The student must be able to apply the various concepts of Computer Vision in other application areas.

Contents:

Digital Image Formation and low-level processing: Overview and State-of-the-art, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing.

Depth estimation and Multi-camera views: Perspective, Binocular Stereopsis: Camera and Epipolar Geometry; Homography, Rectification, DLT, RANSAC, 3-D reconstruction framework; Auto-calibration.

Feature Extraction: Edges-Canny, LOG, DOG; Line detectors (Hough Transform), Corners-Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis-Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean Shift, MRFs, Texture Segmentation; Object detection.

Pattern Analysis: Clustering: K-Means, K-Medoids, Mixture of Gaussians, Classification: Discriminant Function, Supervised, Un-supervised, Semi-supervised; Classifiers: Bayes, KNN, ANN models; Dimensionality Reduction: PCA, LDA, ICA; Non-parametric methods.

Motion Analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.

Shape from X: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges.

Miscellaneous: Applications: CBIR, CBVR, Activity Recognition, computational photography, Biometrics, stitching and document processing; Modern trends - super-resolution; GPU, Augmented Reality; cognitive models, fusion and SR&CS.

Laboratory Work: To implement various techniques and algorithms studied during course.

Course Learning Outcomes: The student will be able to

1. grasp the fundamental problems of computer vision.
2. implement various techniques and algorithms used in computer vision.
3. analyze and evaluate critically the building and integration of computer vision algorithms and systems.
4. demonstrate awareness of the current key research issues in computer vision.

Text/References:

- R. Szeliski, Computer Vision: Algorithms and Applications, Springer-Verlag, 2011).

- A. D. Forsyth and J. Ponce, Computer Vision: A Modern Approach, Pearson Education, 2nd edition, 2012.
- R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2nd Edition, 2003.
- K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, Morgan Kaufmann, 2nd Edition, 1990.
- R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley, 4th edition, 2018.

Evaluation Scheme:

Mid-Semester Examination	25%
End-Semester Examination	45%
Sessionals (Assignments/Quizzes/Lab Evaluation)	30%